**TRƯỜNG ĐẠI HỌC THƯƠNG MẠI**



**Khoa HTTT KT& TMĐT**

**-----🙞🙜🕮🙞🙜-----**

****

**THẢO LUẬN**

**Học Phần**

**KIỂM THỬ PHẦN MỀM**

**Kiểm thử thủ công:** Bài toán quản lý đặt vé máy bay

**Kiểm thử tự động:** Lựa chọn công cụ kiểm thử tự động Selenium IDE thực hiện kiểm thử cho chức năng đăng nhập website bán hàng UNIQLO

GVHD: TS Vũ Diệu Hương

Lớp học phần: 2305ECIT3021

Nhóm thực hiện: Nhóm 3

*Hà Nội, 4 – 2022*

**THÀNH VIÊN NHÓM 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ tên** | **Mã sinh viên** | **Chức vụ** |
| 22 | Ngô Thị Hoài | 20D190076 | Trưởng nhóm |
| 23 | Vũ Thị Hồng | 20D190137 | TV |
| 24 | Nguyễn Liên Huấn | 20D190078 | TV |
| 25 | Trần Bá Hùng | 20D190139 | TV |
| 26 | Trần Quỳnh Hương | 20D190081 | TV |
| 27 | Lê Thị Khuyên | 20D190084 | TV |
| 28 | Nguyễn Thị Loan | 20D190147 | TV |
| 29 | Nguyễn Thị Mai | 20D190029 | TV |
| 30 | Hồ Thị Trà My | 10D190031 | TV |

MỤC LỤC

[**LỜI MỞ ĐẦU** 6](#_Toc134388874)

[**KẾ HOẠCH THỰC HIỆN** 7](#_Toc134388875)

[**PHẦN 1: KIỂM THỬ THỦ CÔNG** 9](#_Toc134388876)

[**1. Mô tả bài toán** 9](#_Toc134388877)

[**1.1. Đặt bài toán** 9](#_Toc134388878)

[**1.2. Biểu đồ quy trình nghiệp vụ của bài toán** 10](#_Toc134388879)

[***1.2.1. Khách hàng lựa chọn vé máy bay*** 10](#_Toc134388880)

[***1.2.2. Khách hàng đặt vé*** 11](#_Toc134388881)

[***1.2.3. Nhân viên thêm mới thông tin chuyến bay*** 11](#_Toc134388882)

[**1.3. Xác định đầu vào, đầu ra** 11](#_Toc134388883)

[***1.3.1. Đầu vào*** 11](#_Toc134388884)

[***1.3.2. Đầu ra*** 13](#_Toc134388885)

[**2. Phân tích** 13](#_Toc134388886)

[***2.1. Yêu cầu chức năng*** 13](#_Toc134388887)

[**2.2. Yêu cầu phi chức năng** 14](#_Toc134388888)

[**2.3. Yêu cầu dữ liệu** 14](#_Toc134388889)

[**2.4. Đặc tả usecase** 15](#_Toc134388890)

[***2.4.1. Lựa chọn vé*** 15](#_Toc134388891)

[***2.4.2. Đặt vé*** 16](#_Toc134388892)

[***2.4.3. Thêm thông tin chuyến bay*** 16](#_Toc134388893)

[***2.4.4. Xem thông tin chuyến bay*** 17](#_Toc134388894)

[**3. Thiết kế và cài đặt chương trình** 18](#_Toc134388895)

[***3.1. Biểu đồ lớp*** 18](#_Toc134388896)

[**3.2. Ngôn ngữ và công cụ lập trình** 18](#_Toc134388897)

[**4. Thiết kế kiểm thử** 19](#_Toc134388898)

[***4.1. Cấu trúc kiểm thử*** 19](#_Toc134388899)

[***4.2. Kỹ thuật đặc tả ca sử dụng*** 22](#_Toc134388900)

[**4.3. Kỹ thuật kiểm thử hộp đen – Black Test** 25](#_Toc134388901)

[***4.3.1. Lớp tương đương*** 25](#_Toc134388902)

[***4.3.2. Giá trị biên*** 27](#_Toc134388903)

[**4.3.3. Bảng quyết định** 30](#_Toc134388904)

[**4.4. Kỹ thuật kiểm thử hộp trắng – While Test** 33](#_Toc134388905)

[***4.3.1. Dòng điều khiển*** 34](#_Toc134388906)

[**4.3.2. Dòng dữ liệu** 37](#_Toc134388907)

[**4.5. Kiểm thử tích hợp** 44](#_Toc134388908)

[**5. Thực nghiệm kiểm thử** 46](#_Toc134388909)

[**5.1. Các ca kiểm thử đã thực hiện version 1** 46](#_Toc134388910)

[***5.1.1. Kiểm thử đặc tả ca sử dụng*** 46](#_Toc134388911)

[***5.1.2. Kỹ thuật lớp tương đương*** 48](#_Toc134388912)

[***5.1.3. Kỹ thuật giá trị biên*** 50](#_Toc134388913)

[***5.1.4. Kỹ thuật bảng quyết định*** 52](#_Toc134388914)

[***5.1.5. Kỹ thuật dòng điều khiển*** 55](#_Toc134388915)

[***5.1.6. Kiểm thử tích hợp*** 55](#_Toc134388916)

[***5.1.7. Nhận xét*** 56](#_Toc134388917)

[**5.2. Kết quả quá trình kiểm thử cuối cùng** 57](#_Toc134388918)

[**6. Đánh giá** 58](#_Toc134388919)

[**1. Giới thiệu phần mềm Selenium** 59](#_Toc134388920)

[**1.1. Các công cụ** 59](#_Toc134388921)

[**1.2. Các tính năng cơ bản của Selenium IDE** 59](#_Toc134388922)

[**1.3. Ưu điểm của công cụ kiểm thử tự động Selenium IDE** 60](#_Toc134388923)

[**1.4. Cách cài đặt Selenium IDE** 61](#_Toc134388924)

[**1.5. Một số thao tác phổ biến** 61](#_Toc134388925)

[**2. Thiết kế kiểm thử cho chức năng đăng nhập của website UNIQLO** 62](#_Toc134388926)

[**2.1. Mô tả các trường hợp thực hiện** 62](#_Toc134388927)

[**2.2. Thực nghiệm các trường hợp** 63](#_Toc134388928)

[**2.3. Đánh giá các trường hợp thực hiện** 67](#_Toc134388929)

[**2.4. Mô tả lại đặc tả use case đăng nhập của website bán hàng UNIQLO** 67](#_Toc134388930)

[**PHẦN 3: PHỤ LỤC** 69](#_Toc134388931)

[**1. Chương trình code** 69](#_Toc134388932)

[**2. Biên bản thảo luận** 88](#_Toc134388933)

[**KẾT LUẬN** 91](#_Toc134388934)

# **LỜI MỞ ĐẦU**

Lỗi phần mềm là một chuyện hiển nhiên trong việc xây dựng các phần mềm hiện nay. Dù những người lập trình viên có cố gắng như thế nào thì thực tế là ngay cả những người xuất sắc nhất cũng không thể tránh khỏi việc đoạn mã mà mình viết ra không có bất kỳ một lỗi nào.

Nhóm 3 chúng em quyết định chọn đề tài:

**Kiểm thử thủ công:** Bài toán quản lý đặt vé máy bay

**Kiểm thử tự động:** Lựa chọn công cụ kiểm thử tự động Selenium IDE thực hiện kiểm thử cho chức năng đăng nhậ website bán hàng UNIQLO

Trong quá trình thực hiện bài thảo luận, do kiến thức và kỹ năng còn hạn chế nên nhóm chúng em không thể tránh khỏi những sai sót, chúng em rất mong nhận được những lời nhận xét, đánh giá cũng như góp ý của cô.

Chúng em xin chân thành cảm ơn!

# **KẾ HOẠCH THỰC HIỆN**

***1. Xác định phương pháp thực hiện***

* Thực hiện tuần tự theo các bước:

(1) Đặt bài toán

(2) Phân tích những chức năng và yêu cầu cần có

(3) Thiết kế cài đặt chương trình

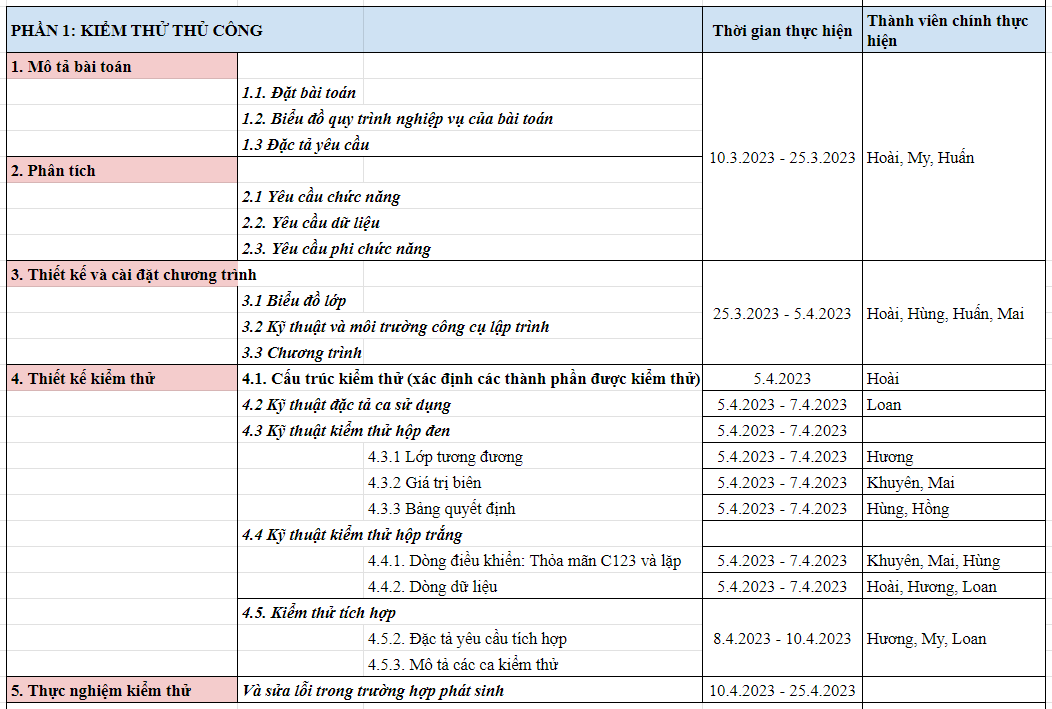
(4) Thiết kế và thực nghiệm kiểm thử

***2. Mục tiêu***

**-** Tạo ra chương trình nhỏ, vừa đáp ứng mô tả phân tích của bài toán, vừa phục vụ cho mục đích thực hành kiểm thử phần mềm.

- Vận dụng linh hoạt các kỹ thuật kiểm thử để phát hiện các lỗi chương trình từ đó khắc phục và sửa chữa, đảm bảo chương trình được vận hành đúng đắn và chính xác.

***3. Chi tiết các giai đoạn thực hiện***



Graphical user interface, text, application, email

Description automatically generated

# **PHẦN 1: KIỂM THỬ THỦ CÔNG**

# **1. Mô tả bài toán**

## **1.1. Đặt bài toán**

Bài toán: Quản lý đặt vé máy bay trực tuyến. Đại lý hiện tại cung cấp dịch vụ đặt vé cho hai chuyến bay bao gồm:

* Thứ nhất, chuyến bay từ sân bay Nội Bài - sân bay Tân Sơn Nhất
* Thứ hai, chuyến bay từ sân bay Nội Bài - sân bay Đà Nẵng

*Quy trình đặt vé được diễn ra như sau:*

Bước 1: Khách hàng muốn thực hiện chuyến bay trước hết cần phải lựa chọn vé máy bay phù hợp với nhu cầu. Đại lý cung cấp các lựa chọn đặt vé máy bay bao gồm lựa chọn thời gian khởi hành, lựa chọn loại vé, loại chuyến bay.

Bước 2: Sau khi đã lựa chọn được loại vé phù hợp thì khách hàng tiến hành cần phải cung cấp những thông tin cá nhân bao gồm: Họ tên, số điện thoại, gmail, ngày sinh.

Bước 3: Sau đó, hệ thống sẽ ghi nhận và dựa trên các thông tin đó để đưa ra giá vé phù hợp. Quy tắc tính giá vé được chi tiết ở mục đầu vào của bài toán.

Bước 4: Khách hàng chọn đặt vé, trong vòng 10 giờ kể từ thời điểm đặt vé, nhân viên của đại lý sẽ gọi điện để xác nhận lại thông tin khách hàng, thông tin đặt vé và thỏa thuận về hình thức thanh toán.

Giả sử, tại một đơn đặt vé, khách hàng chỉ được lựa chọn duy nhất 1 vé để sử dụng.

## **1.2. Biểu đồ quy trình nghiệp vụ của bài toán**

### ***1.2.1. Khách hàng lựa chọn vé máy bay***

Diagram

Description automatically generated

### ***1.2.2. Khách hàng đặt vé***

Diagram

Description automatically generated

### ***1.2.3. Nhân viên thêm mới thông tin chuyến bay***

Diagram, table

Description automatically generated

## **1.3. Xác định đầu vào, đầu ra**

### ***1.3.1. Đầu vào***

Quy tắc xác định vé phụ thuộc vào các biến đầu vào được mô tả như sau:

***a. CB: Chặng bay*:** {Nội Bài – Tân Sơn Nhất, Nội Bài – Đà Nẵng}

Bao gồm 2 chặng bay là:

* Xuất phát từ Nội Bài đến Tân Sơn Nhất: giá khởi điểm là 2.000.000 đồng
* Xuất phát từ Nội Bài đến Đà Nẵng: giá khởi điểm là 1.500.000 đồng

***b. NKH: Ngày khởi hành***

Khách hàng nhập ngày muốn khởi hành chuyến bay, ngày khởi hành được hệ thống chấp nhận là một số nguyên thuộc trong khoảng từ 1 đến 30 ngày sau đó, giá vé được quy định như sau:

* Ngày khởi hành cách từ *[1, 7]* ngày kể từ ngày đặt vé: giá vé sẽ tăng 15% so giá khởi điểm
* Ngày khởi hành cách từ *[8, 15]* ngày kể từ ngày đặt vé: giá vé sẽ biến thiên 0% so với giá khởi điểm
* Ngày khởi hành cách từ *[16, 30]* ngày kể từ ngày đặt vé: giá vé sẽ giảm 15% so với giá khởi điểm
* Nếu ngày khởi hành nhập vào không hợp lệ, hệ thống thông hiển thị thông báo “Ngày khởi hành không hợp lệ!”

*c. LV: Loại vé:* { Phổ thông, Thương Gia, Hạng Nhất}

Khách hàng lựa chọn hạng vé, sau đó giá vé tương ứng với hạng vé được tính như sau:

|  |  |  |
| --- | --- | --- |
| Phổ thông | Thương gia | Hạng nhất |
| Biến thiên 0% so với giá khởi điểm | tăng 20% so với giá khởi điểm | tăng 40% so với giá khởi điểm |

*d. DT: Đối tượng sử dụng vé:* {Người lớn, Trẻ em}

Số tuổi hệ thống tiếp nhận thuộc kiểu số nguyên trong khoảng từ 0 đến 100. Quy tắc xác định giá được mô tả như sau:

* trẻ em (từ 0 tuổi đến 12 tuổi),
* người lớn (từ 13 tuổi trở lên).
* Nếu nhập số tuổi không thuộc trong khoảng đó, hệ thống thông báo “Nhập số tuổi không hợp lệ.”

|  |  |
| --- | --- |
| Người lớn | Trẻ em |
| biến thiên 0% so với giá khởi điểm | giảm 10% so với giá khởi điểm |

*e. LCB: Loại chuyến bay:* {Khứ hồi, Một chiều}

Dựa vào loại chuyến đi mà khách hàng lựa chọn, hệ thống sẽ hiển thị giá vé như sau:

|  |  |
| --- | --- |
| Một chiều | Khứ hồi |
| Biến thiên 0% so với giá khởi điểm | Tăng 80% so với giá khởi điểm |

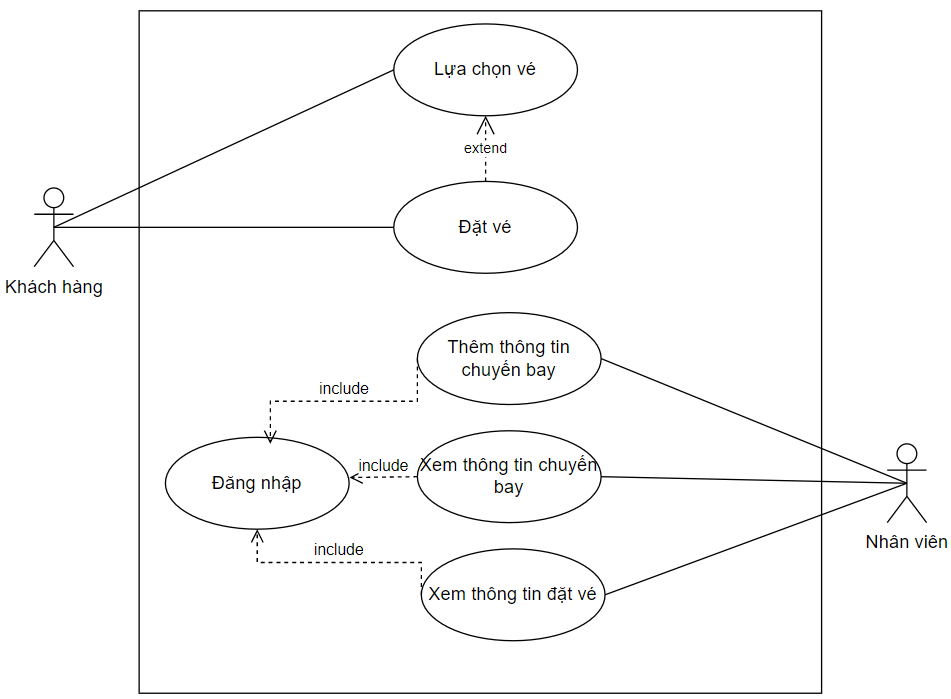
### ***1.3.2. Đầu ra***

*Giá vé máy bay* = Giá khởi điểm (chặng bay) + Giá biến thiên dựa trên ngày khởi hành + Giá biến thiên dựa trên hạng vé + Giá biên thiên dựa trên đối tượng sử dụng vé + Giá vé biến thiên dựa trên loại chuyến bay.

# **2. Phân tích**

## ***2.1. Yêu cầu chức năng***

* Đối với khách hàng, hệ thống cho phép khách hàng thực hiện lựa chọn vé máy bay phù hợp với nhu cầu và đặt vé.
* Đối với nhân viên thêm thông tin chuyến bay, xem thông tin chuyến bay, xem thông tin đặt vé của khách hàng.



## **2.2. Yêu cầu phi chức năng**

* Thông tin khách hàng phải được bảo mật
* Hệ thống giao diện thân thiện, dễ nhìn, dễ sử dụng và phải tương thích với nhiều loại thiết bị và trình duyệt web khác nhau

## **2.3. Yêu cầu dữ liệu**

* Thông tin chuyến bay cần lưu trữ bao gồm: Mã chuyến bay, điểm xuất phát, điểm kết thúc.
* Thông tin về loại vé và hạng ghế.
* Thông tin vé máy bay bao gồm: Mã vé máy bay, loại vé, số ghế ngồi, giá khởi điểm
* Thông tin khách hàng: Mã khách hàng, Họ tên, ngày sinh, số điện thoại, CCCD/CMND

## **2.4. Đặc tả usecase**

### ***2.4.1. Lựa chọn vé***

|  |  |
| --- | --- |
| Use Case Name | Lựa chọn vé |
| Description | Ca sử dụng cho phép khách hàng lựa chọn vé bay theo nhu cầu riêng của mỗi người |
| Actor | Khách hàng |
| Priority | Must have |
| Trigger | Khách hàng phải truy cập hệ thống |
| Pre-Condition(s) | - Tài khoản người dùng đã được phân quyền |
| Post-Condition(s) | - Khách hàng lựa chọn vé thành công  - Hệ thống ghi nhận thông tin lựa chọn vé của khách hàng |
| Basic Flow | 1. Khách hàng click “Lựa chọn vé”.  2. Hệ thống hiển thị thông tin danh sách chuyến bay  3. Khách hàng lựa chọn chuyến bay mong muốn  3.1 Khách hàng chọn chuyến bay mong muốn  3.2 Khách hàng nhập số ngày muốn đặt vé trước  3.4 Khách hàng chọn hạng ghế muốn đặt  3.6 Khách hàng lựa chọn đối tượng sử dụng vé bằng cách nhập độ tuổi của đối tượng  3.7 Khách hàng thực hiện lựa chọn loại vé  4. Hệ thống hiển thị thông tin về vé mà khách hàng vừa lựa chọn |
| Exception Flow | 1. Người dùng không muốn lựa chọn vé và thoát khỏi hệ thống  3.2 Hệ thống hiển thị thông tin thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại số ngày muốn đặt vé trước  3.3 Hệ thống hiển thị thông tin nhập tuổi không hợp lệ và yêu cầu nhập lại số tuổi |

### ***2.4.2. Đặt vé***

|  |  |
| --- | --- |
| Use Case Name | Đặt vé |
| Description | Ca sử dụng cho phép khách hàng đặt vé máy bay theo mong muốn sau khi lựa chọn vé |
| Actor | Khách hàng |
| Priority | Must have |
| Trigger | Khách hàng phải vào website của đại lý cung cấp vé máy bay |
| Pre-Condition(s) | - Khách hàng đã lựa chọn được vé máy bay theo mong muốn |
| Post-Condition(s) | - Khách hàng đặt vé thành công  - Hệ thống ghi nhận thông tin đặt vé của khách hàng |
| Basic Flow | 1. Người dùng click chọn đặt vé  2. Khách hàng điền các thông tin cá nhân cần thiết (Họ tên, ngày sinh, số điện thoại, CCCD/CMND)  3. Hệ thống hiển thị thông tin đặt vé và hiển thị thông báo “Đặt vé thành công”. |
| Exception Flow | 2. Người dùng không muốn đặt vé nữa và thoát ra khỏi hệ thống  5. Tại giao diện đặt vé, khách hàng click “Thoát”  5.1. Hệ thống quy trở lại giao diện nhập thông tin khách hàng. |

### ***2.4.3. Thêm thông tin chuyến bay***

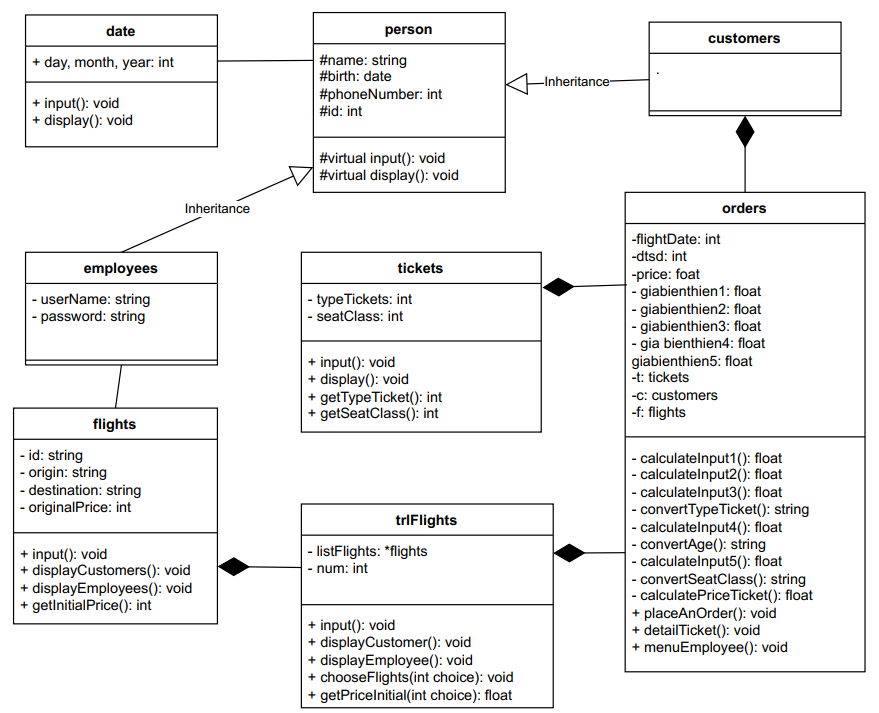
|  |  |
| --- | --- |
| Use Case Name | Thêm thông tin chuyến bay |
| Description | Ca sử dụng cho phép nhân viên thêm thông tin chuyến bay vào hệ thống. |
| Actor | Nhân viên |
| Priority | Must have |
| Trigger | Nhân viên phải đăng nhập vào hệ thống. |
| Pre-Condition(s) | Nhân viên đăng nhập thành công vào hệ thống |
| Post-Condition(s) | * Thêm chuyến bay thành công vào hệ thống. * Hệ thống ghi nhận và cập nhật thông tin lên hệ thống. |
| Basic Flow | 1. Nhân viên đăng nhập và hệ thống đặt vé bằng tài khoản nhân viên. 2. Nhân viên click vào “Thêm thông tin chuyến bay”. 3. Thực hiện nhập thông tin cần thêm. 4. Hệ thống hiển thị thông báo thêm thông tin thành công. |
| Exception Flow | 1. Nhân viên đăng nhập không thành công và yêu cầu đăng nhập lại |

### ***2.4.4. Xem thông tin chuyến bay***

|  |  |
| --- | --- |
| Use Case Name | Thêm thông tin chuyến bay |
| Description | Ca sử dụng cho phép nhân viên thêm thông tin chuyến bay vào hệ thống. |
| Actor | Nhân viên |
| Priority | Must have |
| Trigger | Nhân viên phải đăng nhập vào hệ thống. |
| Pre-Condition(s) | Nhân viên đăng nhập thành công vào hệ thống |
| Post-Condition(s) | * Thêm chuyến bay thành công vào hệ thống. * Hệ thống ghi nhận và cập nhật thông tin lên hệ thống. |
| Basic Flow | 1. Nhân viên đăng nhập và hệ thống đặt vé bằng tài khoản nhân viên. 2. Nhân viên click vào “Thêm thông tin chuyến bay”. 3. Thực hiện nhập thông tin cần thêm. 4. Hệ thống hiển thị thông báo thêm thông tin thành công. |
| Exception Flow | 1. Nhân viên đăng nhập không thành công và yêu cầu đăng nhập lại 2. Chuyến bay nhân viên chọn không tồn tại trên hệ thống.   2.1. Hệ thống hiện thị “Thông tin chuyến bay không tồn tại trên hệ thống!” |

# **3. Thiết kế và cài đặt chương trình**

## ***3.1. Biểu đồ lớp***



## **3.2. Ngôn ngữ và công cụ lập trình**

* Lựa chọn ngôn ngữ C++ để viết chương trình
* Sử dụng môi trường DevC phiên bản 4.9.2 để cài đặt và chạy chương trình.

# **4. Thiết kế kiểm thử**

## ***4.1. Cấu trúc kiểm thử***

Diagram

Description automatically generated

|  |  |
| --- | --- |
| **Kỹ thuật** | **Mục đích** |
| Đặc tả ca sử dụng | * Kiểm thử cho chức năng "Lựa chọn vé". * Vì đây là một trong những chức năng "bắt buộc" của phần mềm, nên cần xác định và kiểm tra hành vi và tương tác của tác nhân "Khách hàng" với hệ thống. * Từ đó đảm bảo hệ thống hoạt động chính xác và đáp ứng kịch bản ca sử dụng |
| Lớp tương đương | * Áp dụng kỹ thuật phân hoạch lớp tương đương để kiểm thử cho chức năng tính giá vé. * Có 5 tham số đầu vào (đã được mô tả trước đó). Trong đó, có 2 giá trị đầu vào thuộc kiểu số nguyên nên có thể dễ dàng chia các giá trị đầu vào thành các lớp tương đương. |
| Giá trị biên | * Mặc dù kỳ thuật phân vùng tương đương hiệu quả trong việc tìm ra các lỗi có trong chương trình, đồng thời giảm số ca kiểm thử cần thực hiện so với kỹ thuật giá trị biên. Tuy nhiên, nó không đảm bảo rẳng sẽ tìm ra các lỗi tiềm ẩn có thể xuất hiện. * Nên áp dụng kỹ thuật giá trị biên để kiểm thử cho chức năng tính giá vé. * Kiểm thử cho các giá trị gần phân vùng tương đương, nhằm tìm ra các lỗi tiềm ẩn gần với giá trị đầu vào. |
| Bảng quyết định | * Áp dụng kỹ thuật bảng quyết định để kiểm thử cho chức năng tính giá vé. * Vì có nhiều điều kiện đầu vào và các giá trị đầu vào có thể kết hợp với nhau một cách phức tạp. * Bảng quyết định là phù hợp để mô tả các giá trị đầu vào và đầu ra tương ứng đối với mỗi trường hợp. |
|  | * Áp dụng kỹ thuật giá trị biên để kiểm thử cho chức năng tính giá vé. * Kiểm thử cho các giá trị gần phân vùng tương đương, nhằm tìm ra các lỗi tiềm ẩn gần với giá trị đầu vào |
|  | * Áp dụng kỹ thuật kiểm thử dòng dữ liệu để kiểm thử chức năng tính giá vé biến thiên dựa trên biến đầu vào "Ngày khởi hành" * Chương trình của hàm này bao gồm 1 vòng lặp, 2 điều kiện rẽ nhánh, * Đường đi khá phức tạp, nên sử dụng kỹ thuật này làm tăng độ bao phủ kiểm thử bằng cách đảm bảo rằng tất cả đường đi trong hàm đều được kiểm tra |
|  | * Áp dụng kỹ thuật kiểm thử dòng dữ liệu để kiểm thử chức năng tính giá vé biến thiên dựa trên biến đầu vào "Ngày khởi hành" * Chương trình của hàm này bao gồm 1 vòng lặp, 2 điều kiện rẽ nhánh, nhận thấy luồng xử lý dữ liệu khá phức tạp. Ngoài ra việc truyền nhận dữ liệu (biến initial Price được truyền từ hàm khác sang hàm này) * Sử dụng kỹ thuật này để kiểm tra và phát hiện các lỗi liên quan đến xử lý dữ liệu. |
|  | * Kiểm thử tích hợp cho hai chức năng:   ----Thêm thông tin chuyến bay  ----Khách hàng tìm thông tin chuyến bay  Và:  ----Thêm thông tin chuyến bay  ----Nhân viên xem thông tin chuyến bay   * Nhằm đảm bảo tương thích giữa 2 thành phần này được tích hợp với nhau. Mục đích nhằm hiển thị chính xác thông tin đối với từng đối tượng |

🡪Mỗi kỹ thuật kiểm thử cũng có những hạn chế và ưu điểm riêng. Do đó, kết hợp các kỹ thuật kiểm thử lại với nhau giúp bổ sung và hoàn thiện những khuyết điểm của từng kỹ thuật kiểm thử để đảm bảo tính toàn diện của quá trình kiểm thử phần mềm.

* **Các miền giá trị được mã hóa lại như sau:**

**Đầu vào:**

1. *Chuyến bay: CB*

1.1. CB1: Nội Bài đến sân bay Tân Sơn Nhất (Giá khởi điểm = 2.000.000)

1.2. CB2: Nội bài đến Đà Nẵng (Giá khởi điểm = 1.500.000)

1. *Ngày khởi hành: NKH*

NKH1: NKH < 1 🡪 Hiển thị thông báo “Nhập ngày khởi hành không hợp lệ!”

NKH2: 1 <= NKH <=7 🡪 Giá vé biến thiên dựa trên NKH = Giá khởi điểm \* 0.15

NKH3: 7< NKH <=15 🡪 Giá vé biến thiên dựa trên NKH = 0

NKH4: 15< NKH <=30 🡪 Giá vé biến thiên dựa trên NKH = - Giá khởi điểm \*0.15

NKH5: NKH > 30 🡪 Hiển thị thông báo “Nhập ngày không hợp lệ!”

1. *Loại vé: LV*

LV1: Phổ thông 🡪 Giá vé biến thiên dựa trên LV = 0

LV2: Thương gia🡪 Giá vé biến thiên dựa trên LV = Giá khởi điểm \* 0.2

LV3: Hạng nhất 🡪 Giá vé biến thiên dựa trên LV = Giá khởi điểm \* 0.4

1. *Đối tượng sử dụng: DT*

DT1: DT <= 0 🡪 Hiển thị thông báo “Nhập số tuổi không hợp lệ!”

DT2: 0 < DT <= 12 🡪 Giá biên thiên dựa trên DT = - Giá khởi điểm \* 0.1

DT3: 12 < DT <= 100 🡪 Giá biên thiên dựa trên DT = 0

DT4: DT > 100 🡪 Hiển thị thông báo “Nhập số tuổi không hợp lệ!”

1. Loại chuyến bay: LCB

LCB1: Khứ hồi 🡪 Giá vé biến thiên dựa trên LCB = Giá khởi điểm \* 0.8

LCB2: Một chiều 🡪 Giá vé biến thiên dựa trên LCB = 0

**Đầu ra:**

GIÁ VÉ = giá khởi điểm + giá biến thiên dựa trên NKH  + giá biến thiên dựa trên LV + giá biến thiên dựa trên DT + giá biến thiên dựa trên LCB

## ***4.2. Kỹ thuật đặc tả ca sử dụng***

***4.2.1. Đặc tả usecase***

Nhóm lựa chọn CSD lựa chọn vé để thiết kế và thực hiện các ca kiểm thử

|  |  |
| --- | --- |
| Use Case Name | Lựa chọn vé |
| Description | Ca sử dụng cho phép khách hàng lựa chọn vé bay theo nhu cầu riêng của mỗi người |
| Actor | Khách hàng |
| Priority | Must have |
| Trigger | Khách hàng phải truy cập hệ thống |
| Pre-Condition(s) | - Tài khoản người dùng đã được phân quyền |
| Post-Condition(s) | - Khách hàng lựa chọn vé thành công  - Hệ thống ghi nhận thông tin lựa chọn vé của khách hàng |
| Basic Flow | 1. Khách hàng click “Lựa chọn vé”.  2. Hệ thống hiển thị thông tin danh sách chuyến bay  3. Khách hàng lựa chọn chuyến bay mong muốn  3.1 Khách hàng chọn chuyến bay mong muốn  3.2 Khách hàng nhập số ngày muốn đặt vé trước  3.4 Khách hàng chọn hạng ghế muốn đặt  3.6 Khách hàng lựa chọn đối tượng sử dụng vé bằng cách nhập độ tuổi của đối tượng  3.7 Khách hàng thực hiện lựa chọn loại vé  4. Hệ thống hiển thị thông tin về vé mà khách hàng vừa lựa chọn |
| Exception Flow | 1. Người dùng không muốn lựa chọn vé và thoát khỏi hệ thống  3.2 Hệ thống hiển thị thông tin thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại số ngày muốn đặt vé trước  3.3 Hệ thống hiển thị thông tin nhập tuổi không hợp lệ và yêu cầu nhập lại số tuổi |

***4.2.2. Thiết kế các ca kiểm thử dựa trên các kịch bản hành vi***

|  |  |
| --- | --- |
|  | **Hành vi** |
| 1 | Khách hàng không muốn chọn vé và thoát khỏi website |
| 2 | Khách hàng thực hiện chọn vé theo từng yếu tố: CB, NKH, LV, DT, LCB 🡪 hiển thị thông tin vé mà khách lựa chọn |
| 3 | Hệ thống hiển thị thông tin thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại số ngày muốn đặt vé trước |
| 4 | Hệ thống hiển thị thông tin nhập tuổi không hợp lệ và yêu cầu nhập lại số tuổi |

***4.2.3. Mô tả các ca kiểm thử điển hình***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC** | **Input** | | | | | **Expected Output** |
| **Chuyến bay (CB)** | **Ngày khởi hành (NKH)** | **Loại vé (LV)** | **Đối tượng sử dụng (DT)** | **Loại chuyến bay (LCB)** |
| 1 | Khách hàng  không muốn chọn vé và thoát khỏi website | | | | | Thoát chương trình |
| 2 | CB1  Nội Bài -Tân Sơn Nhất | 0  NKH1  ( NKH < 1) | LV1  (Phổ thông) | 35  DT3  (Người lớn) | LCB2  (Một chiều) | Hệ thống hiển thị thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại |
| 3 | nt | 4  NKH2  ( 1 <= NKH <=7) | nt | 35  DT3  (Người lớn) | nt | Giá vé = 2.300.000 |
| 4 | nt | 10  NKH3  (7<x<=15) | nt | 35  DT3  (Người lớn) | nt | Giá vé = 2.000.000 |
| 5 | nt | 18  NKH4  (15<x<=30) | nt | 35  DT3  (Người lớn) | nt | Giá vé = 1.700.000 |
| 6 | nt | 38  NKH5  (NKH > 30) | nt | 35  DT3  (Người lớn) | nt | Hệ thống hiển thị thông báo *ngày khởi hành không hợp lệ* và yêu cầu nhập lại |
| 7 | nt | abc | nt | 35  DT3  (Người lớn) | nt | nt |
| 8 | nt | 5  ( 1 <= NKH <=7) | nt | -10  DT1  (DT <= 0 ) | nt | Hệ thống hiển thị thông báo *nhập số tuổi không hợp lệ* và yêu cầu nhập lại |
| 9 | nt | 5  NKH2  ( 1 <= NKH <=7) | nt | 10  DT2  (0<DT<=12)  (Trẻ em) | nt | Giá vé = 2.100.000 |
| 10 | nt | 10  NKH3  (7<x<=15) | nt | 35  DT3  (12 <DT <=100) | nt | Giá vé = 2.000.000 |
| 11 | nt | 10  NKH3  (7<x<=15) | nt | 105  DT4  (DT> 100) | nt | Hệ thống hiển thị thông tin *nhập tuổi không hợp lệ* và yêu cầu nhập lại số tuổi |
| 12 | nt | 18  NKH4  (15<x<=30) | nt | xyz | nt | nt |

## **4.3. Kỹ thuật kiểm thử hộp đen – Black Test**

### ***4.3.1. Lớp tương đương***

***a, Phân lớp tương đương và lựa chọn thành viên***

**Đầu vào:**

*1. Đối với chuyến bay: CB*

1.1. CB1: Nội Bài đến sân bay Tân Sơn Nhất (Giá khởi điểm = 2.000.000)

1.2. CB2: Nội bài đến Đà Nẵng (Giá khởi điểm = 1.500.000)

🡪 Chọn thành viên “CB1”

*2. Đối với ngày khởi hành: NKH*

**A picture containing timeline

Description automatically generated**

🡪Từ đó, xác định được 5 thành viên như sau: -10, 5, 10, 25, 35

*3. Loại vé: LV*

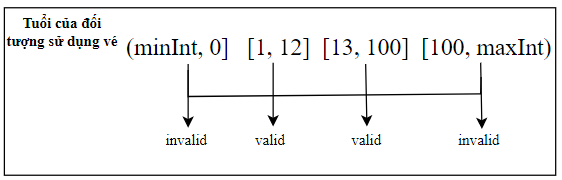
LV1: Phổ thông 🡪 Giá vé biến thiên dựa trên LV = 0

LV2: Thương gia 🡪 Giá vé biến thiên dựa trên LV = Giá khởi điểm \* 0.2

LV3: Hạng nhất 🡪 Giá vé biến thiên dựa trên LV = Giá khởi điểm \* 0.4

🡪Chọn thành viên: LV2: Thương gia

*4. Đối tượng sử dụng: DT*

****

🡪 Từ đó, xác định được 4 thành viên như sau: -12, 6, 26, 126

*5. Loại chuyến bay: LCB*

LCB1: Khứ hồi 🡪 Giá vé biến thiên dựa trên LCB = Giá khởi điểm \* 0.8

LCB2: Một chiều 🡪 Giá vé biến thiên dựa trên LCB = 0

🡪 Chọn thành viên: LCB2: Một chiều

***b, Các trường hợp kiểm thử điển hình***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC** | **Input** | | | | | **Expected Output** |
| **CB** | **NKH** | **LV** | **DT** | **LCB** |
| 1 | Hà Nội – Tân Sơn Nhất | -10 | Thương gia | 6 | Một chiều | Hiển thị thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại |
| 2 | nt | 5 | nt | -12 | nt | Hiển thị thông báo nhập số tuổi không hợp lệ và yêu cầu nhập lại |
| 3 | nt | 5 | nt | 6 | nt | Giá vé: 2.500.000 |
| 4 | nt | 5 | nt | 26 | nt | Giá vé: 2.400.000 |
| 5 | nt | 5 | nt | 126 | nt | Hiển thị thông báo nhập số tuổi không hợp lệ và yêu cầu nhập lại |
| 6 | nt | 10 | nt | -12 | nt | nt |
| 7 | nt | 10 | nt | 6 | nt | Giá vé: 2.200.000 |
| 8 | nt | 10 | nt | 26 | nt | Giá vé: 2.400.000 |
| 9 | nt | 10 | nt | 126 | nt | Hiển thị thông báo nhập số tuổi không hợp lệ và yêu cầu nhập lại |
| 10 | nt | 25 | nt | -12 | nt | nt |
| 11 | nt | 25 | nt | 6 | nt | Giá vé: 1.900.000 |
| 12 | nt | 25 | nt | 26 | nt | Giá vé: 2.100.000 |
| 13 | nt | 25 | nt | 126 | nt | Hiển thị thông báo nhập số tuổi không hợp lệ và yêu cầu nhập lại |
| 14 | nt | 35 | nt | 6 | nt | Hiển thị thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại |

### ***4.3.2. Giá trị biên***

***a, Xác định các giá trị biên cho các biến kiểu số***

* *NKH: Ngày khởi hành:*

**Text

Description automatically generated with medium confidence**

🡪17 giá trị biên cho biến NKH như sau: 0,1,2,4,6,7,8,9,11,14,15,16,17,22,29,30,31

* *Loại vé: LV*

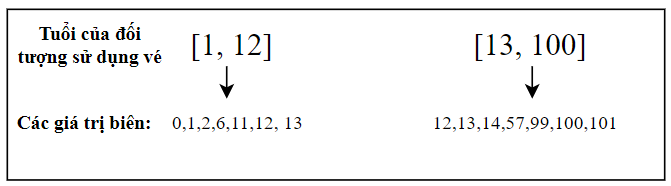
LV1: Phổ thông 🡪 Giá vé biến thiên dựa trên LV = 0

LV2: Thương gia 🡪 Giá vé biến thiên dựa trên LV = Giá khởi điểm \* 0.2

LV3: Hạng nhất 🡪 Giá vé biến thiên dựa trên LV = Giá khởi điểm \* 0.4

🡪Chọn thành viên: LV2: Thương gia

* *DT: Đối tượng sử dụng vé: Độ tuổi (0,100]*

****

🡪11 giá trị biên cho biến DT như sau: 0,1,2,6,11,12,13,14,57,99,100,101

* *Loại chuyến bay: LCB*

LCB1: Khứ hồi 🡪 Giá vé biến thiên dựa trên LCB = Giá khởi điểm \* 0.8

LCB2: Một chiều 🡪 Giá vé biến thiên dựa trên LCB = 0

🡪 Chọn thành viên: LCB2: Một chiều

***b, Các ca kiểm thử***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC** | **Input** | | | | | **Expected Output** |
| **CB** | **NKH** | **LV** | **DT** | **LCB** |
| 1 | Hà Nội – Tân Sơn Nhất | 0 | Thương gia | 10 | Một chiều | Hiển thị thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại |
| 2 | nt | 1 | nt | 10 | nt | Giá vé: 2.500.000 |
| 3 | nt | 2 | nt | 10 | nt | Giá vé: 2.500.000 |
| 4 | nt | 4 | nt | 10 | nt | Giá vé: 2.500.000 |
| 5 | nt | 6 | nt | 10 | nt | Giá vé: 2.500.000 |
| 6 | nt | 7 | nt | 10 | nt | Giá vé: 2.500.000 |
| 7 | nt | 8 | nt | 10 | nt | Giá vé: 2.200.000 |
| 8 | nt | 9 | nt | 10 | nt | Giá vé: 2.200.000 |
| 9 | nt | 11 | nt | 10 | nt | Giá vé: 2.200.000 |
| 10 | nt | 14 | nt | 10 | nt | Giá vé: 2.200.000 |
| 11 | nt | 15 | nt | 10 | nt | Giá vé: 2.400.000 |
| 12 | nt | 16 | nt | 10 | nt | Giá vé: 1.900.000 |
| 13 | nt | 17 | nt | 10 | nt | Giá vé: 1.900.000 |
| 14 | nt | 22 | nt | 10 | nt | Giá vé: 1.900.000 |
| 15 | nt | 29 | nt | 10 | nt | Giá vé: 1.900.000 |
| 16 | nt | 30 | nt | 10 | nt | Giá vé: 1.900.000 |
| 17 | nt | 31 | nt | 10 | nt | Hiển thị thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại |
| 18 | nt | 10 | nt | 0 | nt | Hiển thị thông báo nhập số tuổi không hợp lệ và yêu cầu nhập lại |
| 19 | nt | 10 | nt | 1 | nt | Giá vé: 2.200.000 |
| 20 | nt | 10 | nt | 2 | nt | Giá vé: 2.200.000 |
| 21 | nt | 10 | nt | 6 | nt | Giá vé: 2.200.000 |
| 22 | nt | 10 | nt | 11 | nt | Giá vé: 2.200.000 |
| 23 | nt | 10 | nt | 12 | nt | Giá vé: 2.200.000 |
| 24 | nt | 10 | nt | 13 | nt | Giá vé: 2.400.000 |
| 25 | nt | 10 | nt | 14 | nt | Giá vé: 2.400.000 |
| 25 | nt | 10 | nt | 57 | nt | Giá vé: 2.400.000 |
| 27 | nt | 10 | nt | 99 | nt | Giá vé: 2.400.000 |
| 28 | nt | 10 | nt | 100 | nt | Giá vé: 2.400.000 |
| 29 | nt | 10 | nt | 101 | nt | Hiển thị thông báo nhập số tuổi không hợp lệ và yêu cầu nhập lại. |

### **4.3.3. Bảng quyết định**

***a, Dựa vào đặc tả không gian đầu vào ta xác định được các phân hoạch sau:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên biến** | **Ký hiệu** | **Điều kiện** | **Giá trị đại diện** |
| 1 | Chặng bay | CB | | |
| 2 |  | CB1 | Nội Bài đến sân bay Tân Sơn Nhất |  |
| 3 |  | CB2 | Nội Bài đến sân bay Đà Nẵng |  |
| 4 | Ngày khởi hành | NKH | | |
| 5 |  | NKH1 | NKH<1 | -1 |
| 6 |  | NKH2 | 1<=NKH<=7 | 6 |
| 7 |  | NKH3 | 7<NKH<=15 | 10 |
| 8 |  | NKH4 | 15<NKH<=30 | 20 |
| 9 |  | NKH5 | NKH>30 | 31 |
| 10 | Loại vé | LV |  |  |
| 11 |  | LV1 | Phổ thông |  |
| 12 |  | LV2 | Thương gia |  |
| 13 |  | LV3 | Hạng nhất |  |
| 14 | Đối tượng sử dụng vé | DT | | |
| 15 |  | DT1 | DT<=0 | -1 |
| 16 |  | DT2 | 0<DT<=12 | 6 |
| 17 |  | DT3 | 12<DT<=100 | 20 |
| 18 |  | DT4 | 100<DT | 101 |
| 19 | Loại chuyến bay | LCB |  |  |
| 20 |  | LCB1 | Khứ hồi |  |
| 21 |  | LCB2 | một chiều |  |

Lưu ý, đối với các biến kiểu string, lựa chọn 1 giá trị đại diện (là những ô được tô màu) khi tiến hành kiểm thử thiết kế các ca kiểm thử.

***b, Xác định hành động***

|  |  |
| --- | --- |
| STT | Hành động |
| A1 | Thông báo: Nhập số tuổi không hợp lệ |
| A2 | Thông báo: Nhập ngày khởi hành không hợp lệ |
| A3 | Thông báo hợp lệ, tính giá cho khách hàng |

***🡪 Bảng quyết định cho việc đặt vé máy bay:***

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **R1** | **R2** | **R3** | **R4** | **R5** | **R6** | **R7** | **R8** | **R9** | **R10** |
| **Điều kiện** | CB1 | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** |
| CB2 | F | F | F | F | F | F | F | F | F | F |
| NKH1 | **T** | **T** | **T** | **T** | F | F | F | F | F | F |
| NKH2 | F | F | F | F | **T** | **T** | **T** | **T** | F | F |
| NKH3 | F | F | F | F | F | F | F | F | **T** | **T** |
| NKH4 | F | F | F | F | F | F | F | F | F | F |
| NKH5 | F | F | F | F | F | F | F | F | F | F |
| LV1 | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** |
| LV2 | F | F | F | F | F | F | F | F | F | F |
| LV3 | F | F | F | F | F | F | F | F | F | F |
| DT1 | **T** | F | F | F | **T** | F | F | F | **T** | F |
| DT2 | F | **T** | F | F | F | **T** | F | F | F | **T** |
| DT3 | F | F | **T** | F | F | F | **T** | F | F | F |
| DT4 | F | F | F | **T** | F | F | F | **T** | F | F |
| LCB1 | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** |
| LCB2 | F | F | F | F | F | F | F | F | F | F |
| **Hành động** | A1 | X |  |  | X | X |  |  | X | X |  |
| A2 | X | X | X | X |  |  |  |  |  |  |
| A3 |  |  |  |  |  | X | X |  |  | X |
| **Kết quả** | Giá | Lỗi | Lỗi | Lỗi | Lỗi | Lỗi | 3700000 | 3900000 | Lỗi | Lỗi | 3400000 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **R11** | **R12** | **R13** | **R14** | **R15** | **R16** | **R17** | **R18** | **R19** | **R20** |
| **Điều kiện** | CB1 | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** |
| CB2 | F | F | F | F | F | F | F | F | F | F |
| NKH1 | F | F | F | F | F | F | F | F | F | F |
| NKH2 | F | F | F | F | F | F | F | F | F | F |
| NKH3 | **T** | **T** | F | F | F | F | F | F | F | F |
| NKH4 | F | F | **T** | **T** | **T** | **T** | F | F | F | F |
| NKH5 | F | F | F | F | F | F | **T** | **T** | **T** | **T** |
| LV1 | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** |
| LV2 | F | F | F | F | F | F | F | F | F | F |
| LV3 | F | F | F | F | F | F | F | F | F | F |
| DT1 | F | F | **T** | F | F | F | **T** | F | F | F |
| DT2 | F | F | F | T | F | F | F | **T** | F | F |
| DT3 | **T** | F | F | F | **T** | F | F | F | **T** | F |
| DT4 | F | **T** | F | F | F | **T** | F | F | F | **T** |
| LCB1 | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** | **T** |
| LCB2 | F | F | F | F | F | F | F | F | F | F |
| **Hành động** | A1 |  | X | X |  |  | X | X | X | X | X |
| A2 |  |  |  |  |  |  | X |  |  | X |
| A3 | X |  |  | X | X |  |  |  |  |  |
| **Kết quả** | Giá | 3600000 | Lỗi | Lỗi | 3100000 | 3300000 | Lỗi | Lỗi | Lỗi | Lỗi | Lỗi |

## **4.4. Kỹ thuật kiểm thử hộp trắng – While Test**

**Sử dụng kỹ thuật kiểm thử hộp trắng cho hàm caculateInput2():**

Mục đích của hàm này: Tính giá biến thiên dựa trên biến đầu vào số ngày khởi hành trước, được mô tả lại như sau:

Khách hàng nhập ngày muốn khởi hành chuyến bay, ngày khởi hành được hệ thống chấp nhận là một số nguyên thuộc trong khoảng từ 1 đến 30 ngày sau đó, giá vé được quy định như sau:

* Ngày khởi hành cách từ *[1, 7]* ngày kể từ ngày đặt vé: giá vé sẽ tăng 15% so giá khởi điểm
* Ngày khởi hành cách từ *[8, 15]* ngày kể từ ngày đặt vé: giá vé sẽ biến thiên 0% so với giá khởi điểm
* Ngày khởi hành cách từ *[16, 30]* ngày kể từ ngày đặt vé: giá vé sẽ giảm 15% so với giá khởi điểm
* Nếu ngày khởi hành nhập vào không hợp lệ, hệ thống thông hiển thị thông báo “Ngày khởi hành không hợp lệ!”

*Chương trình như sau:*

Text

Description automatically generated

### ***4.3.1. Dòng điều khiển***

***a, Độ bao phủ C1, C2***

* *Sơ đồ dòng điều khiển với độ bao phủ C1 và C2*

*Diagram, schematic

Description automatically generated*

* *Xác định độ phức tạp Cyclomatic:*

- C = P + 1 = 3 + 1 = 4

- Đồ thị trên có 3 nút quyết định nhị phân nên cần xác định 4 đường đi để đảm bảo tất cả các câu lệnh của chức năng được thực thi ít nhất một lần.

* *Sinh ca kiểm thử:*

|  |  |  |  |
| --- | --- | --- | --- |
| **TC** | **Đường đi** | **Input tương ứng**  **(flightDate;initialPrice)** | **Expected Output**  **(giabienthien2)** |
| 1 | 1, 2S, 8Đ, 9 | (5;1.500.000) | 225.000 |
| 2 | 1, 2S, 8S, 10Đ, 11 | (11;2.000.000) | 0 |
| 3 | 1, 2S, 8S, 10S, 12 | (20;1.500.000) | -225.000 |
| 4 | 1, 2Đ, 3, 4, 5, 6, 7, 2S, 8Đ, 9 | (0;2.000.000),  (4;2.000.000) | 300.000 |

→ Nhận thấy với cả 4 ca kiểm thử trên, chúng ta mới có thể đạt được độ phủ cho cả 2 độ đo C1 và C2.

***b, Độ bao phủ C3***

* *Sơ đồ dòng điều khiển với độ bao phủ C3:*

Diagram, schematic

Description automatically generated

* *Xác định độ phức tạp Cyclomatic:*

- Đồ thị trên có 5 nút quyết định nhị phân nên có độ phức tạp C = P + 1 = 5 + 1 = 6

- Đồ thị trên có 5 nút quyết định nhị phân nên cần xác định 6 đường đi để đảm bảo tất cả các câu lệnh của chức năng được thực thi ít nhất một lần.

* *Sinh ca kiểm thử:*

|  |  |  |  |
| --- | --- | --- | --- |
| **TC** | **Đường đi** | **Input tương ứng**  **(flightDate;initialPrice)** | **Expected Output**  **(giabienthien2)** |
| 1 | 1, 2.1S, 2.2S, 2.3S, 8Đ, 9 | (4;2.000.000) | 300.000 |
| 2 | 1, 2.1S, 2.2S, 2.3S, 8S, 10Đ,11 | (12;1.500.000) | 0 |
| 3 | 1, 2.1S, 2.2S, 2.3S, 8S, 10S,12 | (25;2.000.000) | -300.000 |
| 4 | 1, 2.1Đ, 3, 4, 5, 6, 7, 2.1S, 2.2S, 2.3S, 8Đ ,9 | (0;1.500.000);  (2;1.500.000) | 225.000 |
| 5 | 1, 2.1S, 2.2Đ, 3, 4, 5, 6, 7, 2.1S, 2.2S, 2.3S, 8S, 10Đ, 11 | (31;2.000.000);  (15;2.000.000) | 0 |
| 6 | 1, 2.1S, 2.2S, 2.3Đ, 3, 4, 5, 6, 7, 2.1S, 2.2S, 2.3S, 8S, 10S, 12 | (###,1.500.000);  (28,1.500.000) | -225.000 |

🡪Nhận thấy, với cả 6 ca kiểm thử trên, chúng ta có thể đạt được độ phủ cho độ đo C3.

***c, Lặp***

Nhóm tiến hành kiểm thử vòng lặp đối với câu lệnh while từ dòng lệnh số 5 – 11

* Xác định các trường hợp kiểm thử: Với vòng lặp trên, số lần lặp tối đa không thể xác định trước số vòng lặp tối đa. Các trường hợp kiểm thử bao gồm:

- Thực hiện 0 lần

- Thực hiện 1 lần

- Thực hiện 2 lần

- Thực hiện k lần, 2 < k < n - 1, với n là số lần lặp tối đa của vòng lặp 🡪 chọn k = 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC | Số lần lặp | Inputs flightDate | EO | RO | Note |
| 1 | 0 | 0 | Lỗi |  |  |
| 2 | 1 | 31; 1 | Giá vé tăng 15% so với giá khởi điểm |  |  |
| 3 | 2 | 0; 31; 10 | Giá vé giữ nguyên so với giá khởi điểm |  |  |
| 4 | 3 | 0; 31; abc; 20 | Giá vé giảm 15% so với giá khởi điểm |  |  |

### **4.3.2. Dòng dữ liệu**

*Sử dụng sơ đồ chuyển trạng thái chương trình ứng với mỗi biến để  tìm ra trạng thái “bất thường” mà không theo chuẩn lập trình. Những bất thường, thường liên quan đến việc khai báo, khởi tạo giá trị cho biến và sử dụng chúng.*

Các thành phần của sơ đồ chuyển trạng thái của một chương trình tương ứng với mỗi biến bao gồm:

Các trạng thái gồm:

-U: biến chưa được gán giá trị

-D: biến được gán giá trị nhưng chưa được sử dụng

-R: biến được sử dụng

-A: trạng thái lỗi

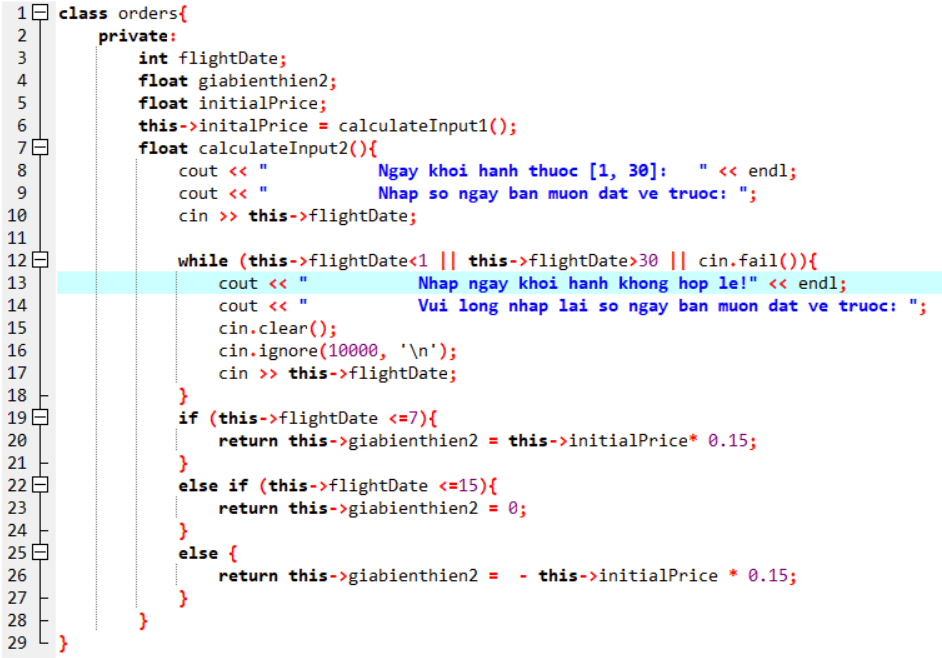
Các hành động bao gồm:

-d: biến được gán giá trị

-r: biến được sử dụng

-u: biến chưa được gán giá trị hoặc được khai báo lại và chưa được gán giá trị

*Nhóm sử dụng hàm tính giá vé biến thiên dựa trên số ngày khởi hành chuyến bay trước:*



***a, Đồ thị dòng dữ liệu cho tổng thể các biến***



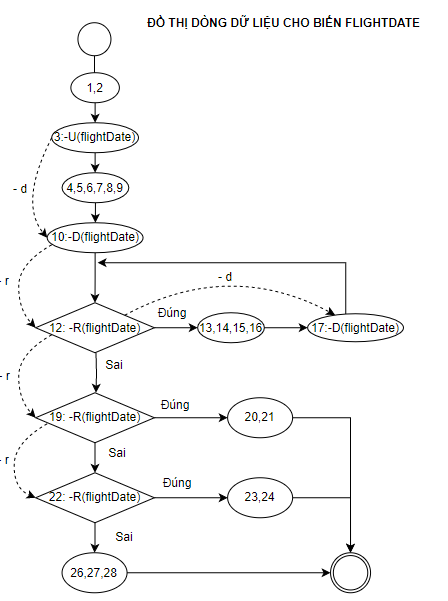
Đồ thị có 3 nút quyết định nhị phân nên có độ phức tạp: C = P + 1 = 3 +1 = 4

* Xác định các đường đi:

|  |  |
| --- | --- |
| **TC** | **Đường đi** |
| 1 | 12345678910, 12S, 19Đ, 20 |
| 2 | 12345678910, 12S, 19S, 22Đ, 23 2S, 8S, 10Đ, 11 |
| 3 | 12345678910, 12S, 19S, 22S, 26 |
| 4 | 12345678910, 12Đ, 13-14-15-16-17, 12S,  19Đ, 20 |

***b, Kiểm thử đời sống cho biến flightDate***

* *Sơ đồ chuyển trạng thái cho biến flightDate*



* *Mô tả sơ đồ chuyển trạng thái*

- Dòng lệnh 3, ban đầu biến được khai báo và chưa được gán trị nên trạng thái tại đây là U.

- Cho đến dòng lệnh số 10, gặp câu lệnh nhập vào giá trị cho biến flightDate (hành động d) nên trạng thái của biến được chuyển thành D.

- Biến này được sử dụng ở dòng lệnh số 12 (hành động r) nên tại đây, trạng thái của biến được chuyển thành R. Tại dòng lệnh số 12 sẽ có trường hợp tương ứng với 2 hành động xảy ra:

*# Nếu điều kiện 12 là đúng → Trường hợp 1*

*# Nếu điều kiện 12 là sai → Trường hợp 2*

*# Trường hợp 1:*

- Xuất hiện hành động gán lại giá trị cho biến ở dòng lệnh số 17 (hành động) nên trạng thái của biến quay trở về D.

- Sau đó, lại quay lại hành động r ở dòng lệnh, cứ lặp lại như vậy nếu điều kiện 12 sai.

*# Trường hợp 2:*

- Biến vẫn tiếp tục được sử ở câu lệnh 19 nên trạng thái của biến vẫn không thay đổi

- Tại dòng lệnh thứ 22, biến vẫn tiếp tục được sử dụng, như vậy trạng thái của biến vẫn giữ nguyên R.

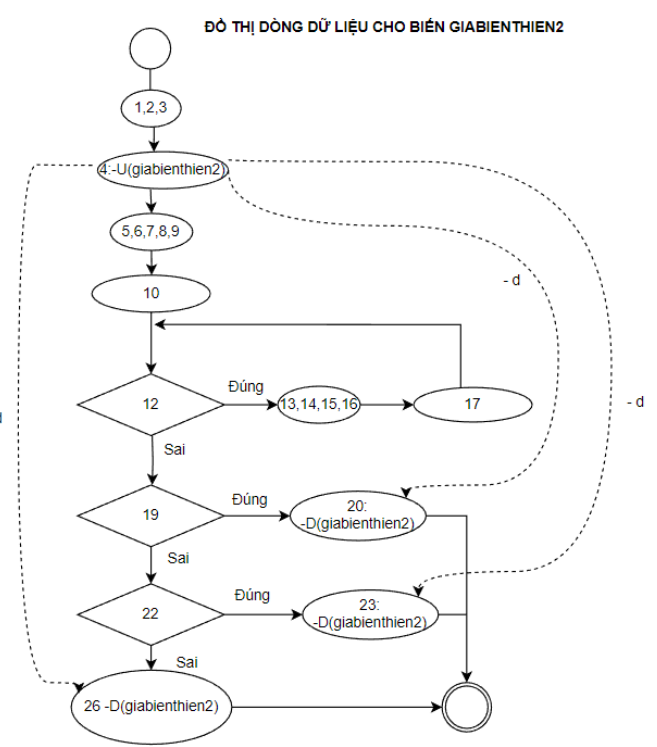
* *Xác định các kịch bản bình thường và bất thường*

|  |  |  |  |
| --- | --- | --- | --- |
| **TC** | **Đường đi** | **Lần lượt trạng thái flightDate** | **Đánh giá** |
| 1 | 12345678910, 12S, 19Đ, 20 | -U-D-R | Bình thường |
| 2 | 12345678910, 12S, 19S, 22Đ, 23 2S, 8S, 10Đ, 11 | -U-D-R-R-R | Bình thường |
| 3 | 12345678910, 12S, 19S, 22S, 26 | -U-D-R-R-R | Bình thường |
| 4 | 12345678910, 12Đ, 13-14-15-16-17, 12S, 19Đ, 20 | -U-D-R-D-R-R | Bình thường |

→ Kết luận: Trong 5 kịch bản trên, không chứa trạng thái lỗi, bất thường của biến flightDate

***b, Kiểm thử đời sống cho biến giabienthien2***

* *Sơ đồ chuyển trạng thái cho biến giabienthien2*

**

* *Mô tả sơ đồ chuyển trạng thái*

- Dòng lệnh 4, ban đầu biến được khai báo và chưa được gán trị nên trạng thái tại đây là U.

- Cho đến dòng lệnh số 20, 23, 26, gặp câu lệnh nhập vào giá trị cho biến giabienthien2  (hành động d) nên trạng thái của biến được chuyển thành D.

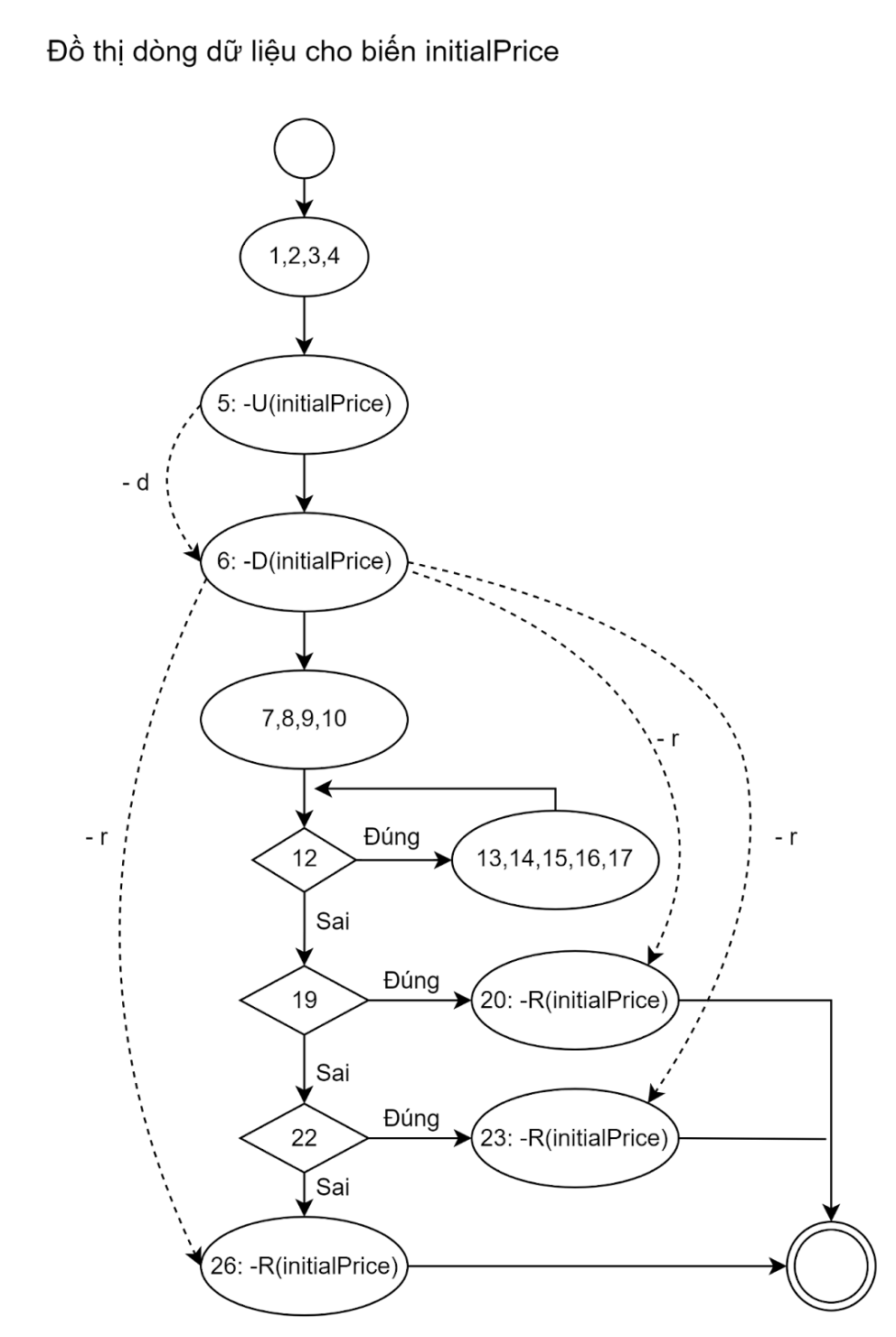
* *Xác định các kịch bản bình thường và bất thường*

|  |  |  |  |
| --- | --- | --- | --- |
| **TC** | **Đường đi** | **Lần lượt trạng thái của biến** | **Đánh giá** |
| 1 | 12345678910, 12S, 19S, 22S, 26 | -U-D | bình thường |
| 2 | 12345678910, 12S, 19Đ, 20 | -U-D | bình thường |
| 3 | 12345678910, 12S, 19S, 22Đ, 2323 | -U-D | bình thường |
| 4 | 12345678910, 12Đ, 13-14-15-16-17, 12S,  19Đ, 20 | -U-D | bình thường |

🡪Trong các kịch bản trên đều không chứa hoạt động nào bất thuờng cả.

*c, Kiểm thử đời sống cho biến initialPrice*

* *Sơ đồ chuyển trạng thái cho bến initialPrice*



* *Mô tả sơ đồ chuyển trạng thái*

- Dòng lệnh 5, ban đầu biến được khai báo và chưa được gán trị nên trạng thái tại đây là U.

- Đến dòng lệnh số 6, gặp câu lệnh nhập vào giá trị cho biến initialPrice (hành động d) nên trạng thái của biến được chuyển thành D.

- Biến này được sử dụng ở dòng lệnh số 20 (hành động r), nên tại đây trạng thái của biến được chuyển thành R

- Tương tự như vậy, ở dòng thứ 23 và 26 trạng thái của biến là R

* *Xác định các kịch bản bình thường và bất thường*

|  |  |  |  |
| --- | --- | --- | --- |
| **TC** | **Đường đi** | **Lần lượt trạng thái** | **Đánh giá** |
| 1 | 12345678910, 12S, 19Đ, 20 | -U-D-R | Bình thường |
| 2 | 12345678910, 12S, 19S, 22Đ, 23 2S, 8S, 10Đ, 11 | -U-D-R | Bình thường |
| 3 | 12345678910, 12S, 19S, 22S, 26 | -U-D-R | Bình thường |
| 4 | 12345678910, 12Đ, 13-14-15-16-17, 12S, 19Đ, 20 | -U-D-R | Bình thường |

→ Kết luận: Trong 4 kịch bản trên, không chứa trạng thái lỗi, bất thường của biến initialPrice

## **4.5. Kiểm thử tích hợp**

Tích hợp 2 module thêm thông tin vé và khách hàng xem thông tin chuyến bay

***a, Đặc tả yêu cầu tích hợp***

* Lựa chọn kiểm thử tích hợp cho 2 module của chương trình:
* Nhân viên thêm thông tin vé
* Khách hàng xem thông tin chuyến bay
* Thông tin vé máy bay được hệ thống quản lý bao gồm: Mã chuyến bay, điểm xuất phát, điểm đến, và giá khởi điểm.

→Tuy nhiên với vai trò là khách hàng truy cập vào hệ thống, được phép xem các thông tin về chuyến bay sau:

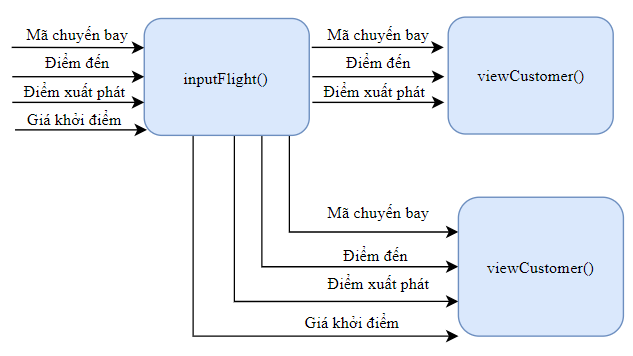
* Mã chuyến bay
* Điểm xuất phát
* Điểm đến

→Với vai trò là nhân viên, được phép xem các thông tin về chuyến bay bao gồm:

* Mã chuyến bay
* Điểm xuất phát
* Điểm đến
* Giá khởi điểm

→ Mục đích kiểm thử: Đảm bảo rằng thông chuyến bay mà nhân viên nhập vào được hiển thị chính xác và đối với từng đối tượng người dùng khác nhau.

*🡪 Mô tả các biến đầu vào và đầu ra cho hàm như sau:*



***b, Mô tả ca kiểm thử***

* Trường hợp tích hợp chức năng nhập thông tin chuyến bay và hiển thị thông tin chuyến bay ra màn hình với vai trò nhân viên:

|  |  |  |
| --- | --- | --- |
| **TC** | **Input** | **Expected output** |
| 1 | Số lượng chuyến bay 2:   * Thông tin chuyến bay 1:   Mã chuyến bay: 1  Điểm khởi hành: Hà Nội  Điểm đến: Tân Sơn Nhất  Giá khởi điểm: 2.000.000   * Thông tin chuyến bay 2:   Mã chuyến bay: 2  Điểm khởi hành: Hà Nội  Điểm đến: Đà Nẵng  Giá khởi điểm: 1.500.000 | Số lượng chuyến bay 2:   * Thông tin chuyến bay 1:   Mã chuyến bay: 1  Điểm khởi hành: Hà Nội  Điểm đến: Tân Sơn Nhất  Giá khởi điểm: 2.000.000   * Thông tin chuyến bay 2:   Mã chuyến bay: 2  Điểm khởi hành: Hà Nội  Điểm đến: Đà Nẵng  Giá khởi điểm: 1.500.000 |

* Trường hợp tích hợp chức năng nhập thông tin chuyến bay và hiển thị thông tin chuyến bay ra màn hình với vai trò khách hàng:

|  |  |  |
| --- | --- | --- |
| **TC** | **Input** | **Expected output** |
| 1 | Số lượng chuyến bay 2:   * Thông tin chuyến bay 1:   Mã chuyến bay: 1  Điểm khởi hành: Hà Nội  Điểm đến: Tân Sơn Nhất  Giá khởi điểm: 2.000.000   * Thông tin chuyến bay 2:   Mã chuyến bay: 2  Điểm khởi hành: Hà Nội  Điểm đến: Đà Nẵng  Giá khởi điểm: 1.500.000 | Số lượng chuyến bay 2:   * Thông tin chuyến bay 1:   Mã chuyến bay: 1  Điểm khởi hành: Hà Nội  Điểm đến: Tân Sơn Nhất   * Thông tin chuyến bay 2:   Mã chuyến bay: 2  Điểm khởi hành: Hà Nội  Điểm đến: Đà Nẵng |

# **5. Thực nghiệm kiểm thử**

## **5.1. Các ca kiểm thử đã thực hiện version 1**

### ***5.1.1. Kiểm thử đặc tả ca sử dụng***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TC** | **Input** | | | | | **Expected Output** |  | **Result** |
| **CB** | **nkh** | **LV** | **DT** | **LCB** | **Actual output** |
| 1 | Khách hàng  không muốn chọn vé và thoát khỏi website | | | | | Thoát chương trình | Hiển thị nháy không ngừng | fail |
| 2 | HN  -  TSN | 0 | Phổ thông | 35 | Một chiều | Hệ thống hiển thị thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại |  | pass |
| 3 | nt | 4 | nt | 35 | nt | Giá vé = 2.300.000 |  | pass |
| 4 | nt | 10 | nt | 35 | nt | Giá vé = 2.000.000 |  | pass |
| 5 | nt | 18 | nt | 35 | nt | Giá vé = 1.700.000 |  | pass |
| 6 | nt | 38 | nt | 35 | nt | Hệ thống hiển thị thông báo *ngày khởi hành không hợp lệ* và yêu cầu nhập lại |  | pass |
| 7 | nt | abc | nt | 35 | nt | nt | Hiển thị nháy không ngừng | fail |
| 8 | nt | 5 | nt | -10 | nt | Hệ thống hiển thị thông báo *nhập số tuổi không hợp lệ* và yêu cầu nhập lại |  | pass |
| 9 | nt | 5 | nt | 10 | nt | Giá vé = 2.100.000 |  | pass |
| 10 | nt | 10 | nt | 35 | nt | Giá vé = 2.000.000 |  | pass |
| 11 | nt | 10 | nt | 105 | nt | Hệ thống hiển thị thông tin *nhập tuổi không hợp lệ* và yêu cầu nhập lại số tuổi |  | pass |
| 12 | nt | 18 | nt | xyz | nt | nt | Hiển thị nháy không ngừng | fail |

🡪 Ca kiểm thử phát hiện lỗi: Số 7 và 12

* Xác định lỗi thực tế: Hệ thống hiển thị nháy do vòng lặp không kết thúc.
* Khoanh vùng lỗi: Lỗi phát sinh ở biến đầu vào là độ tuổi và ngày khởi hành (khi nhập vào là kí tự string)
* Nguyên nhân: Chưa giải quyết được trường hợp đầu vào các biến không phải dạng số

### ***5.1.2. Kỹ thuật lớp tương đương***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TC** | **Input** | | | | | **Expected Output** | **Actual output** | **Result** |
| **CB** | **NKH** | **LV** | **DT** | **LCB** |  |  |  |
| 1 | Hà Nội – Tân Sơn Nhất | -10 | Thương gia | 6 | Một chiều | Hiển thị thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại | Hiển thị giá vé biến thiên là 0 | fail |
| 2 | nt | 5 | nt | -12 | nt | Hiển thị thông báo nhập số tuổi không hợp lệ và yêu cầu nhập lại | Hiển thị giá vé biến thiên là 0 | fail |
| 3 | nt | 5 | nt | 6 | nt | Giá vé: 2.500.000 |  | pass |
| 4 | nt | 5 | nt | 26 | nt | Giá vé: 2.400.000 |  | pass |
| 5 | nt | 5 | nt | 126 | nt | Hiển thị thông báo nhập số tuổi không hợp lệ và yêu cầu nhập lại |  | pass |
| 6 | nt | 10 | nt | -12 | nt | nt | Hiển thị giá vé biến thiên là 0 | fail |
| 7 | nt | 10 | nt | 6 | nt | Giá vé: 2.200.000 |  | pass |
| 8 | nt | 10 | nt | 26 | nt | Giá vé: 2.400.000 |  | pass |
| 9 | nt | 10 | nt | 126 | nt | Hiển thị thông báo nhập số tuổi không hợp lệ và yêu cầu nhập lại |  | pass |
| 10 | nt | 25 | nt | -12 | nt | nt | Hiển thị giá vé biến thiên là 0 | fail |
| 11 | nt | 25 | nt | 6 | nt | Giá vé: 1.900.000 |  | pass |
| 12 | nt | 25 | nt | 26 | nt | Giá vé: 2.100.000 |  | pass |
| 13 | nt | 25 | nt | 126 | nt | Hiển thị thông báo nhập số tuổi không hợp lệ và yêu cầu nhập lại |  | pass |
| 14 | nt | 35 | nt | 6 | nt | Hiển thị thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại |  | pass |

🡪 Ca kiểm thử phát hiện lỗi: Số 1, 2, 6, 10

* Khoanh vùng lỗi: Lỗi phát sinh ở biến đầu vào là độ tuổi và ngày khởi hành (khi nhập vào là số nhỏ hơn 0)
* Nguyên nhân: Chưa xét điều kiện hợp lệ để tính giá vé trường hợp ngày khởi hành và độ tuổi nhập vào là một số nhỏ hơn 0.

### ***5.1.3. Kỹ thuật giá trị biên***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TC** | **Input** | | | | | **Expected Output** | **Actual output** | **Result** |
| **CB** | **NKH** | **LV** | **DT** | **LCB** |  |
| 1 | Hà Nội – Tân Sơn Nhất | 0 | Thương gia | 10 | Một chiều | Hiển thị thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại | Hiển thị giá vé biến thiên là 0 | fail |
| 2 | nt | 1 | nt | 10 | nt | Giá vé: 2.500.000 |  | pass |
| 3 | nt | 2 | nt | 10 | nt | Giá vé: 2.500.000 |  | pass |
| 4 | nt | 4 | nt | 10 | nt | Giá vé: 2.500.000 |  | pass |
| 5 | nt | 6 | nt | 10 | nt | Giá vé: 2.500.000 |  | pass |
| 6 | nt | 7 | nt | 10 | nt | Giá vé: 2.500.000 |  | pass |
| 7 | nt | 8 | nt | 10 | nt | Giá vé: 2.200.000 |  | pass |
| 8 | nt | 9 | nt | 10 | nt | Giá vé: 2.200.000 |  | pass |
| 9 | nt | 11 | nt | 10 | nt | Giá vé: 2.200.000 |  | pass |
| 10 | nt | 14 | nt | 10 | nt | Giá vé: 2.200.000 |  | pass |
| 11 | nt | 15 | nt | 10 | nt | Giá vé: 2.400.000 |  | pass |
| 12 | nt | 16 | nt | 10 | nt | Giá vé: 1.900.000 |  | pass |
| 13 | nt | 17 | nt | 10 | nt | Giá vé: 1.900.000 |  | pass |
| 14 | nt | 22 | nt | 10 | nt | Giá vé: 1.900.000 |  | pass |
| 15 | nt | 29 | nt | 10 | nt | Giá vé: 1.900.000 |  | pass |
| 16 | nt | 30 | nt | 10 | nt | Giá vé: 1.900.000 |  | pass |
| 17 | nt | 31 | nt | 10 | nt | Hiển thị thông báo ngày khởi hành không hợp lệ và yêu cầu nhập lại |  | pass |
| 18 | nt | 10 | LV2 | 0 | nt | Hiển thị thông báo nhập số tuổi không hợp lệ và yêu cầu nhập lại |  | pass |
| 19 | nt | 10 | nt | 1 | nt | Giá vé: 2.200.000 |  | pass |
| 20 | nt | 10 | nt | 2 | nt | Giá vé: 2.200.000 |  | pass |
| 21 | nt | 10 | nt | 6 | nt | Giá vé: 2.200.000 |  | pass |
| 22 | nt | 10 | nt | 11 | nt | Giá vé: 2.200.000 |  | pass |
| 23 | nt | 10 | nt | 12 | nt | Giá vé: 2.200.000 |  | pass |
| 24 | nt | 10 | nt | 13 | nt | Giá vé: 2.400.000 |  | pass |
| 25 | nt | 10 | nt | 57 | nt | Giá vé: 2.400.000 |  | pass |
| 25 | nt | 10 | nt | 99 | nt | Giá vé: 2.400.000 |  | pass |
| 27 | nt | 10 | nt | 100 | nt | Giá vé: 2.400.000 |  | pass |
| 28 | nt | 10 | nt | 101 | nt | Hiển thị thông báo nhập số tuổi không hợp lệ và yêu cầu nhập lại. | Hiển thị giá vé biến thiên là 0 | pass |

🡪 Ca kiểm thử phát hiện lỗi: Số 1, 17, 18, 28

* Khoanh vùng lỗi: Lỗi phát sinh ở biến đầu vào là độ tuổi và ngày khởi hành (khi nhập vào là số thuộc giá trị biên mạnh)
* Nguyên nhân: Chưa xét điều kiện hợp lệ để tính giá vé trường hợp ngày khởi hành nhập vào là số thuộc biên mạnh (n-1 và n+1)

### ***5.1.4. Kỹ thuật bảng quyết định***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case** | **Input** | | | | | **Expected Output** | **Actual output** | **Result** |
| **CB** | **NKH** | **LV** | **DT** | **LCB** |
| 1 | Nội bài – Tân Sơn Nhất | -1 | Phổ thông | -1 | Một chiều | Thông báo: Nhập số tuổi không hợp lệ  Thông báo: Nhập ngày khởi hành không hợp lệ  Yêu cầu nhập lại | Hệ thống hiện thị giá biến thiên = 0 | Fail |
| 2 | NT | -1 | NT | 6 | NT | Thông báo: Nhập ngày khởi hành không hợp lệ  Yêu cầu nhập lại ngày khởi hành | Nt | Fail |
| 3 | NT | -1 | NT | 20 | NT | Thông báo: Nhập ngày khởi hành không hợp lệ  Yêu cầu nhập lại ngày khởi hành | Nt | Fail |
| 4 | NT | -1 | NT | 101 | NT | Thông báo: Nhập số tuổi không hợp lệ  Thông báo: Nhập ngày khởi hành không hợp lệ  Yêu cầu nhập lại | Nt | Fail |
| 5 | NT | 6 | NT | -1 | NT | Thông báo: Nhập số tuổi không hợp lệ  Yêu cầu nhập lại số tuổi | Nt | Fail |
| 6 | NT | 6 | NT | 6 | NT | Tính giá vé=3.700.000 |  | Pass |
| 7 | NT | 6 | NT | 20 | NT | Tính giá vé=3.900.000 |  | Pass |
| 8 | NT | 6 | NT | 101 | NT | Thông báo: Nhập số tuổi không hợp lệ  Yêu cầu nhập lại số tuổi | Nt | Fail |
| 9 | NT | 10 | NT | -1 | NT | Thông báo: Nhập số tuổi không hợp lệ  Yêu cầu nhập lại số tuổi | Nt | Fail |
| 10 | NT | 10 | NT | 6 | NT | Tính giá vé=3.400.000 |  | Pass |
| 11 | NT | 10 | NT | 20 | NT | Tính giá vé=3.600.000 |  | Pass |
| 12 | NT | 10 | NT | 101 | NT | Thông báo: Nhập số tuổi không hợp lệ  Yêu cầu nhập lại số tuổi | nt | Fail |
| 13 | NT | 20 | NT | -1 | NT | Thông báo: Nhập số tuổi không hợp lệ  Yêu cầu nhập lại số tuổi | Nt | Fail |
| 14 | NT | 20 | NT | 6 | NT | Tính giá vé=3.100.000 |  | Pass |
| 15 | NT | 20 | NT | 20 | NT | Tính giá vé=3.300.000 |  | Pass |
| 16 | NT | 20 | NT | 101 | NT | Thông báo: Nhập số tuổi không hợp lệ  Yêu cầu nhập lại số tuổi | nt | Fail |
| 17 | NT | 31 | NT | -1 | NT | Thông báo: Nhập số tuổi không hợp lệ  Thông báo: Nhập ngày khởi hành không hợp lệ  Yêu cầu nhập lại | Nt | Fail |
| 18 | NT | 31 | NT | 6 | NT | Thông báo: Nhập ngày khởi hành không hợp lệ  Yêu cầu nhập lại ngày khởi hành | nt | Fail |
| 19 | NT | 31 | NT | 20 | NT | Thông báo: Nhập ngày khởi hành không hợp lệ  Yêu cầu nhập lại ngày khởi hành | nt | Fail |
| 20 | NT | 31 | NT | 101 | NT | Thông báo: Nhập số tuổi không hợp lệ  Thông báo: Nhập ngày khởi hành không hợp lệ  Yêu cầu nhập lại | Nt | Fail |

🡪 Ca kiểm thử phát hiện lỗi: Số 1, 2, 3, 4, 5, 8, 9, 12, 13, 16, 17, 18, 19, 20

* Khoanh vùng lỗi: Lỗi phát sinh ở biến đầu vào:

+ Ngày khởi hành: NKH < 1

+ Độ tuổi: DT<=0

* Nguyên nhân: Chưa xét điều kiện hợp lệ để tính giá vé trong trường hợp trên.

### ***5.1.5. Kỹ thuật dòng điều khiển***

Đối với kiểm thử dòng điều khiển với độ bao phủ C123, và vòng lặp, sau khi thực thi, kết quả hệ thống đều cho ra giống kết quả dự kiến.

### ***5.1.6. Kiểm thử tích hợp***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC** | **Input** | **Expected output** | **Actual output** | **Results** |
| 1 | Số lượng chuyến bay 2:   * Thông tin chuyến bay 1:   Mã chuyến bay: 1  Điểm khởi hành: Hà Nội  Điểm đến: Tân Sơn Nhất  Giá khởi điểm: 2.000.000   * Thông tin chuyến bay 2:   Mã chuyến bay: 2  Điểm khởi hành: Hà Nội  Điểm đến: Đà Nẵng  Giá khởi điểm: 1.500.000 | Số lượng chuyến bay 2:   * Thông tin chuyến bay 1:   Mã chuyến bay: 1  Điểm khởi hành: Hà Nội  Điểm đến: Tân Sơn Nhất  Giá khởi điểm: 2.000.000   * Thông tin chuyến bay 2:   Mã chuyến bay: 2  Điểm khởi hành: Hà Nội  Điểm đến: Đà Nẵng  Giá khởi điểm: 1.500.000 | Số lượng chuyến bay 2:   * Thông tin chuyến bay 1:   Mã chuyến bay: 1  Điểm khởi hành: Hà Nội  Điểm đến: Tân Sơn Nhất  Giá khởi điểm: 2.000.000   * Thông tin chuyến bay 2:   Mã chuyến bay: 2  Điểm khởi hành: Hà Nội  Điểm đến: Đà Nẵng  Giá khởi điểm: 1.500.000 | Pass |

* Trường hợp tích hợp chức năng nhập thông tin chuyến bay và hiển thị thông tin chuyến bay ra màn hình với vai trò khách hàng:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC** | **Input** | **Expected output** | **Actual output** | **Results** |
| 1 | Số lượng chuyến bay 2:   * Thông tin chuyến bay 1:   Mã chuyến bay: 1  Điểm khởi hành: Hà Nội  Điểm đến: Tân Sơn Nhất   * Thông tin chuyến bay 2:   Mã chuyến bay: 2  Điểm khởi hành: Hà Nội  Điểm đến: Đà Nẵng | Số lượng chuyến bay 2:   * Thông tin chuyến bay 1:   Mã chuyến bay: 1  Điểm khởi hành: Hà Nội  Điểm đến: Tân Sơn Nhất   * Thông tin chuyến bay 2:   Mã chuyến bay: 2  Điểm khởi hành: Hà Nội  Điểm đến: Đà Nẵng | Số lượng chuyến bay 2:   * Thông tin chuyến bay 1:   Mã chuyến bay: 1  Điểm khởi hành: Hà Nội  Điểm đến: Tân Sơn Nhất   * Thông tin chuyến bay 2:   Mã chuyến bay: 2  Điểm khởi hành: Hà Nội  Điểm đến: Đà Nẵng | Pass |

### ***5.1.7. Nhận xét***

Qua thực nghiệm kiểm thử version 1, các trường hợp phát sinh lỗi đến từ đầu vào NKH (ngày khởi hành) và đối tượng sử dụng vé (DT). Bảng tổng hợp lỗi:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **stt** | **Kỹ thuật** | **Chức năng** | **Số Testcase** | **Số TC passed** | **Số TC failed** | **Ghi chú** |
| 1 | Đặc tả ca sử dụng | Lựa chọn vé | 12 | 9 | 3 | Lỗi xuất hiện ở đầu vào nhập vào là ký tự string |
| 2 | Lớp tương đương | nt | 14 | 10 | 4 | Lỗi xuất hiện ở phân hoạch tương tương đương NKH < 1 và DT <= 1 |
| 3 | Giá trị biên | nt | 29 | 25 | 4 | Lỗi xuất hiện ở giá trị biên mạnh và biên yếu của 2 biến đầu vào NKH và DT |
| 4 | Bảng quyết định | nt | 20 | 6 | 14 | Lỗi phát hiện ở đầu vào NKH < 1 và DT < 1 |

🡪Tiến hành sửa lại chương trình, và kiểm thử version2, kết quả các ca kiểm thử đều pass**.**

## **5.2. Kết quả quá trình kiểm thử cuối cùng**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **stt** | **Kỹ thuật** | **Chức năng** | **Số Testcase** | **Số TC passed** | **Số TC failed** | **Ghi chú** |
| 1 | Đặc tả ca sử dụng | Lựa chọn vé | 12 | 12 | 0 | Chương trình đã vượt qua tất cả các testcase của bộ kiểm thử đề ra. |
| 2 | Lớp tương đương | nt | 14 | 14 | 0 | Chương trình đã vượt qua tất cả các testcase của bộ kiểm thử đề ra. |
| 3 | Giá trị biên | nt | 28 | 28 | 0 | Chương trình đã vượt qua tất cả các testcase của bộ kiểm thử đề ra. |
| 4 | Bảng quyết định | nt | 20 | 20 | 0 | Chương trình đã vượt qua tất cả các testcase của bộ kiểm thử đề ra. |
| 5 | Độ bao phủ C1 | Tính giá biên thiên dựa trên số ngày khởi hành trước | 4 | 4 | 0 |  |
| 6 | Độ bao phủ C2 | Nt | 4 | 4 | 0 |  |
| 7 | Độ bao phủ vòng lặp | Nt | 4 | 4 | 0 |  |
| 8 | Kiểm thử đời sống cho biến flightDate | Nt | 4 | - |  | Bình thường |
| 9 | Kiểm thử đời sống cho biến giabienthien2 | Nt | 4 | - |  | Bình thường |
| 10 | Kiểm thử đời sống cho biến initialPrice | Nt | 4 | - |  | Bình thường |

# **6. Đánh giá**

* Tất cả các test case đã có dữ liệu kiểm thử, đầu vào, kết quả thực tế và kết quả mong đợi.
* Tất cả các test case đều có nguồn gốc từ các yêu cầu của dự án (đã được cung cấp đầy đủ trong yêu cầu của phần mềm)
* Đã có đầy đủ các kịch bản tích cực và tiêu cực.
* Các test case đều có các bước về cách thực hiện kiểm thử, đều không quá phức tạp, đảm bảo yếu tố đơn giản, dễ hiểu.
* Mô cả các testcase dễ hiểu.
* Không còn các trường hợp kiểm thử trùng lặp, loại bỏ test case không mong muốn để tránh dư thừa.

**PHẦN 2: KIỂM THỬ TỰ ĐỘNG**

Nhóm lựa chọn công cụ kiếm thử tự động Selenium IDE để trải nghiệm chức năng đăng nhập của website bán hàng UNIQNO <https://www.uniqlo.com/vn/vi> . Tuy nhiên, do không có bản đặc tả yêu cầu về chức năng này, nên để phục vụ mục đích cho bài kiểm thử, nhóm sẽ tự mô tả các trường hợp sẽ thực hiện tự động trên hệ thống. Qua quá trình trải nghiệm website, nhóm viết lại đặc tả use case của chức năng đăng nhập website này.

# **1. Giới thiệu phần mềm Selenium**

## **1.1. Các công cụ**

Selenium là một công cụ kiểm thử phần mềm tự động mã nguồn mở được sử dụng để kiểm thử các ứng dụng web. Selenium được sử dụng để tạo và chạy các kịch bản kiểm thử trên trình duyệt web và giúp đơn giản hóa quá trình kiểm thử tự động.

Selenium cung cấp một số công cụ để tạo các kịch bản kiểm thử, bao gồm**:**

**(1)** Selenium IDE (Integrated Development Environment)**:** là một công cụ giao diện người dùng đơn giản để tạo kịch bản kiểm thử mà không cần biết nhiều về lập trình.

**(2)** Selenium WebDriver**:** là một API để tương tác với trình duyệt web và thực hiện các hành động trên trang web

**(3)** Selenium Grid**:** Selenium Grid là một công cụ để chạy các kịch bản kiểm thử trên nhiều trình duyệt và thiết bị khác nhau.

Selenium hỗ trợ nhiều ngôn ngữ lập trình như Java, Python, C#, Ruby, JavaScript và nhiều ngôn ngữ khác nữa. Nó cũng có thể tích hợp với các công cụ kiểm thử khác như TestNG, Junit, Nunit và nhiều công cụ khác.

Selenium là một công cụ kiểm thử tự động mạnh mẽ và đáng tin cậy cho kiểm thử ứng dụng web, và được sử dụng rộng rãi trong ngành công nghiệp phần mềm.

## **1.2. Các tính năng cơ bản của Selenium IDE**

Selenium IDE là một công cụ kiểm thử tự động được sử dụng để ghi lại và thực thi các kịch bản kiểm thử trên trình duyệt web. Các tính năng cơ bản của Selenium IDE bao gồm:

1. Ghi lại và thực thi kịch bản kiểm thử: Selenium IDE cho phép bạn ghi lại các hành động trên trình duyệt web và thực thi chúng để kiểm tra tính năng của trang web.
2. Tự động hoá các hành động kiểm thử: Selenium IDE có thể tự động thực thi các kịch bản kiểm thử đã được tạo để kiểm tra các chức năng trên trang web.
3. Xem trực tiếp các hành động được ghi lại: Selenium IDE cung cấp chế độ xem trực tiếp để bạn có thể xem các hành động được ghi lại khi thực hiện kiểm thử.
4. Sửa đổi và tùy chỉnh các hành động kiểm thử: Selenium IDE cho phép bạn chỉnh sửa và tùy chỉnh các hành động kiểm thử theo ý muốn.
5. Lưu trữ các kịch bản kiểm thử: Selenium IDE cho phép bạn lưu trữ các kịch bản kiểm thử trong các tệp tin để sử dụng lại trong tương lai.
6. Hỗ trợ nhiều ngôn ngữ lập trình: Selenium IDE hỗ trợ các ngôn ngữ lập trình như Java, C#, Python, Ruby, và PHP để giúp bạn tùy chỉnh và mở rộng các hành động kiểm thử.
7. Kiểm thử đa nền tảng: Selenium IDE hỗ trợ kiểm thử trên nhiều nền tảng, bao gồm Windows, Mac và Linux.

## **1.3. Ưu điểm của công cụ kiểm thử tự động Selenium IDE**

|  |  |
| --- | --- |
| **Ưu điểm** | **Hạn chế** |
| - Dễ sử dụng: Selenium IDE là một công cụ giao diện người dùng đơn giản và dễ sử dụng cho người mới bắt đầu với kiểm thử tự động.  - Tạo kịch bản kiểm thử dễ dàng: Selenium IDE cho phép bạn ghi lại các hành động trên trình duyệt web và tạo kịch bản kiểm thử tự động chỉ với vài thao tác đơn giản.  - Tự động thực thi kiểm thử: Selenium IDE có thể tự động thực thi các kịch bản kiểm thử đã được tạo để kiểm tra các chức năng trên trang web.  - Phát triển nhanh chóng: Vì Selenium IDE là một công cụ đơn giản và dễ sử dụng, bạn có thể phát triển các kịch bản kiểm thử nhanh chóng để kiểm tra các tính năng mới trên trang web. | - Hạn chế tính linh hoạt: Selenium IDE chỉ hỗ trợ một số trình duyệt và không hỗ trợ một số tính năng phức tạp hơn của kiểm thử tự động.  - Không thích hợp cho kiểm thử lớn: Selenium IDE không phù hợp cho kiểm thử tự động trên quy mô lớn hoặc kiểm thử tự động tự động hóa.  - Không hỗ trợ các ngôn ngữ lập trình: Selenium IDE không hỗ trợ các ngôn ngữ lập trình khác nhau và không cho phép bạn tùy chỉnh các hành động kiểm thử theo ý muốn. |

## **1.4. Cách cài đặt Selenium IDE**

Các bước để tải Selenium IDE trên trình duyệt Chorme, chúng ta làm theo các bước sau đây:

1. Mở trình duyệt Chrome và truy cập vào trang web chính thức của Selenium IDE tại địa chỉ: <https://www.selenium.dev/selenium-ide/>
2. Nhấp vào nút “Download” trên trang web này.
3. Chọn phiên bản phù hợp với trình duyệt Chrome và hệ điều hành.
4. Sau khi tải xuống, mở trình duyệt Chrome và nhấp vào biểu tượng ba chấm ở góc trên cùng bên phải của trình duyệt.
5. Chọn “More Tools” và sau đó chọn “Extensions”.
6. Nhấp vào nút “Load unpacked” và chọn thư mục mà bạn đã giải nén từ bước 3.
7. Selenium IDE đã được cài đặt thành công trên trình duyệt Chrome

## **1.5. Một số thao tác phổ biến**

Một vài thao tác phổ biến được sử dụng trong cộng cụ Selenium IDE như sau:

|  |  |
| --- | --- |
| Tên thao tác | Hành động |
| open | Mở một trang Web cụ thể |
| click | Bấm vào một đối tượng trên trang Web |
| type | Nhập dữ liệu vào một trường trên trang Web |
| assert | Kiểm tra xem một điều kiện nào đó có đúng hay không |
| verify | Để kiểm tra liên tục điều kiện nào đó đúng hay không, thực hiện cho đến khi điều kiện đúng |
| waitFor | Đợi cho đến khi một trạng thái cụ thể nào được đạt trên trang Web |
| store | Lưu giá trị của một biến vào một biến khác |
| gotolf | Chuyển hướng đến một nhãn dán nhất định nếu một điều kiện được đáp ứng |

# **2. Thiết kế kiểm thử cho chức năng đăng nhập của website UNIQLO**

## **2.1. Mô tả các trường hợp thực hiện**

***2.1.1. Xác định các hành vi mong đợi***

|  |  |
| --- | --- |
|  | Những hành vi kỳ vọng xảy ra |
| 1 | Hiển thị giao diện tìm kiếm sản phẩm |
| 2 | Hiển thị thông báo “Vui lòng nhập mật khẩu” |
| 3 | Hiển thị thông báo “Vui lòng nhập tên đăng nhập và mật khẩu hợp lê” |
| 4 | Hiển thị đồng thời thông báo “Vui lòng nhập tên đăng nhập” và “Vui lòng nhập mật khẩu”. |
| 5 | Hiển thị thông báo “Tài khoản không tồn tại, vui lòng kiểm tra lại”. |

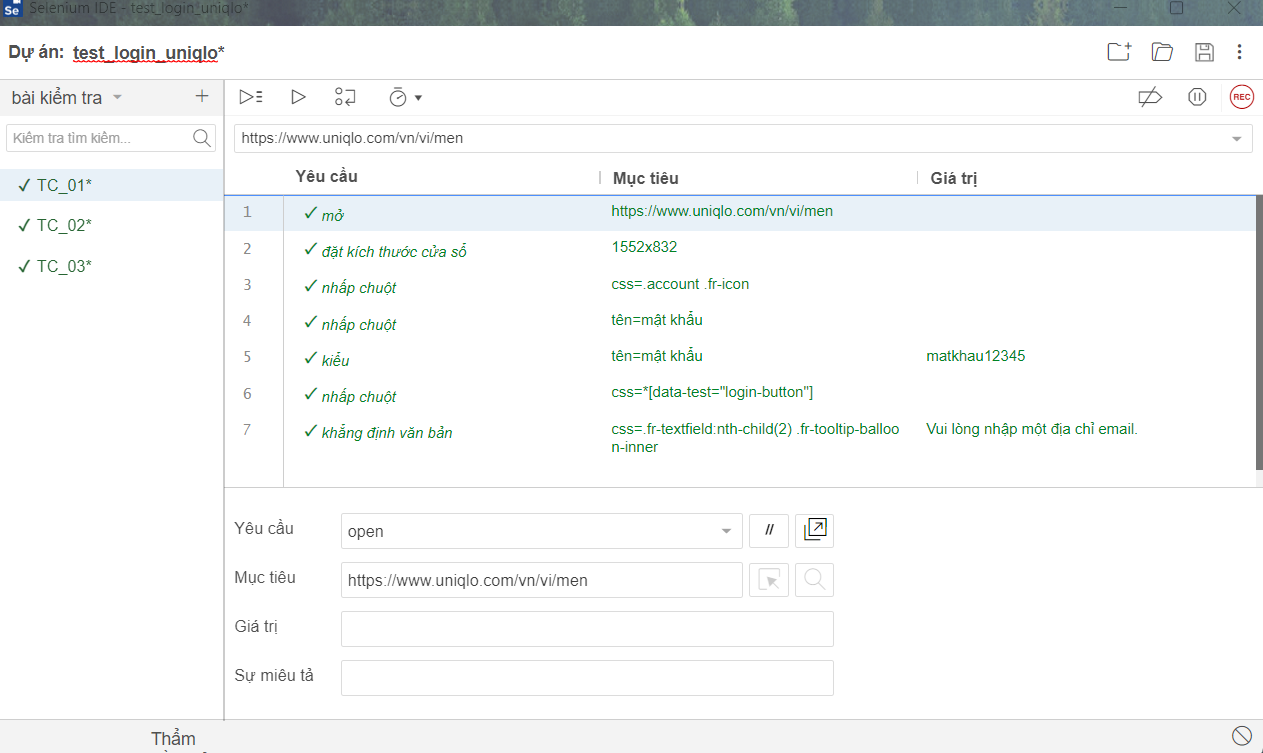
***2.1.2. Mô tả các trường hợp thực hiện và đánh giá***

|  |  |  |  |
| --- | --- | --- | --- |
| **TC** | **Đầu vào** | **Đầu ra** | **Expected output** |
| TC\_01 | Email: empty  Mật khẩu: matkhau12345 | Từ chối quyền truy cập | Chưa nhập email |
| TC\_02 | Email: chuoinguyen895@gmail.com  Mật khẩu: empty | Từ chối quyền truy cập | Chưa nhập mật khẩu |
| TC\_03 | Email: chuoinguyen895@gmail.com  Mật khẩu: matkhau12345 | Cho phép truy cập vào | Chuyển sang trang chủ của uniqlo.com |
| TC\_04 | Email: empty  Mật khẩu: empty | Từ chối quyền truy cập | Chưa nhập email |
| Chưa nhập mật khẩu |

## **2.2. Thực nghiệm các trường hợp**

Minh họa thực hiện các ca kiểm thử như sau:

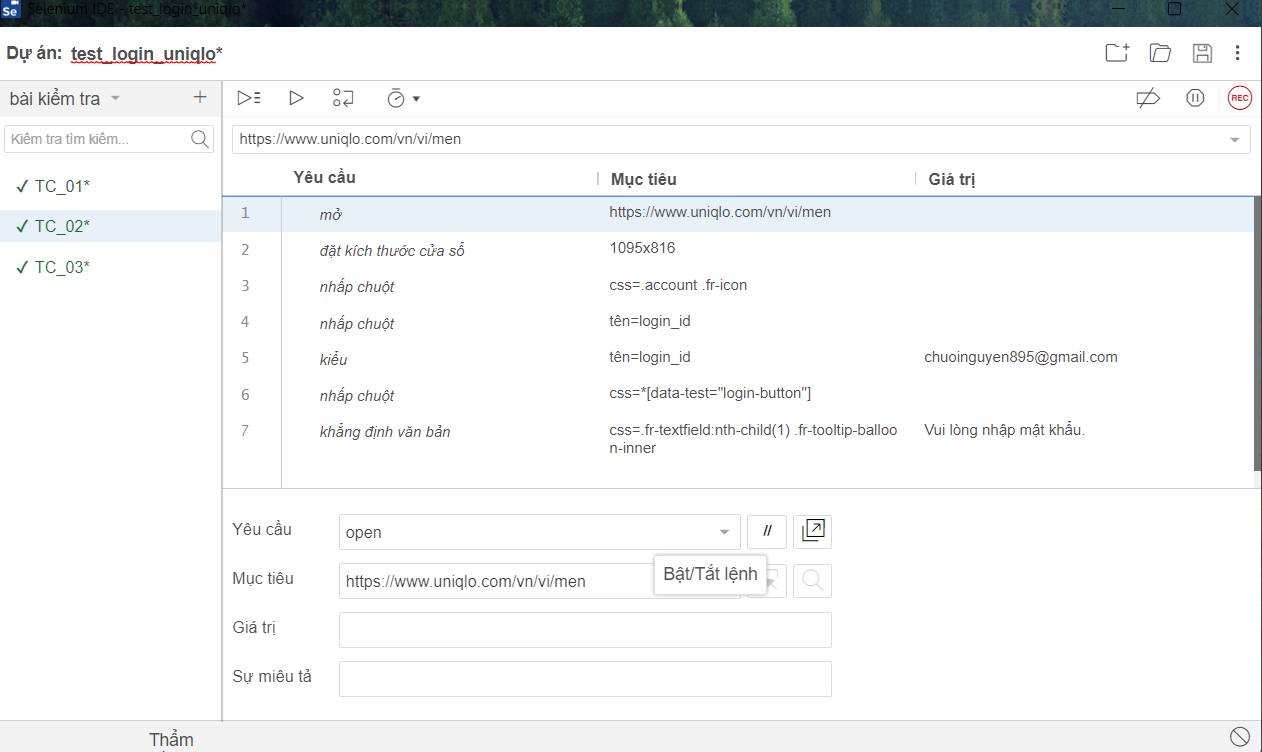
**a, TC\_01: Bỏ trống trường email**

****

*Các bước thực hiện:*

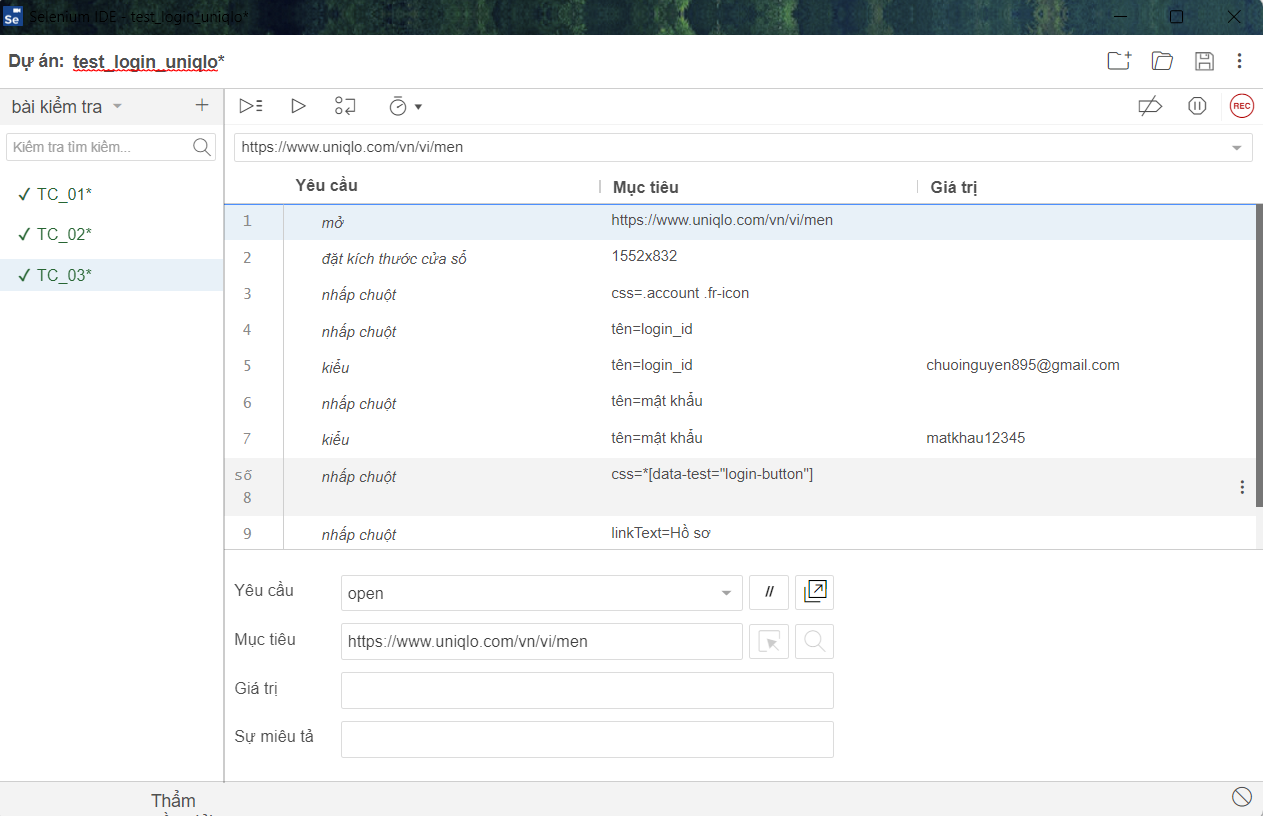
* B1: Mở trang web uniqlo.com
* B2: Đặt kích thước cửa sổ (Chọn biểu tượng dẫn đến trang đăng nhập)
* B3: Nhấp chuột vào form đăng nhập
* B4: Nhấp vào ô nhập liệu mật khẩu
* B5: Nhập mật khẩu
* Nhập mật khẩu với giá trị là: matkhau12345
* B6: Nhấp vào nút Đăng nhập
* B7: Xác minh lỗi bằng Khẳng định văn bản (assert text)
* Sau khi nhấn nút đăng nhập, để kiểm tra xem có hiển thị lỗi không thì dùng assert text để kiểm tra. Kiểm tra xem có hiển thị lỗi dạng văn bản là “Vui lòng nhập một địa chỉ email.” Hay không, nếu có là xác minh lỗi thành công.

**B, TC\_02: Bỏ trống trường mật khẩu**

****

* B1: Mở trang web uniqlo.com
* B2: Đặt kích thước cửa sổ (Dẫn đến trang đăng nhập)
* B3: Nhấp chuột vào form đăng nhập
* B4: Nhấp vào ô nhập liệu email
* B5: Nhập tài khoản email
* Nhập email với giá trị là: chuoinguyen895@gmail.com
* B6: Nhấp vào nút Đăng nhập
* B7: Xác minh lỗi bằng Khẳng định văn bản (assert text)
* Sau khi nhấn nút đăng nhập, để kiểm tra xem có hiển thị lỗi không thì dùng assert text để kiểm tra. Kiểm tra xem có hiển thị lỗi dạng văn bản là “Vui lòng nhập mật khẩu.” hay không, nếu có là xác minh lỗi thành công.

**C, TC\_03: Điền đủ và chính xác email, mật khẩu**

****

* B1: Mở trang web uniqlo.com
* B2: Đặt kích thước cửa sổ (Chọn biểu tượng dẫn đến trang đăng nhập)
* B3: Nhấp chuột vào form đăng nhập
* B4: Nhấp vào ô nhập liệu email
* B5: Nhập tài khoản email
* Nhập tài khoản email là: chuoinguyen895@gmail.com
* B6: Nhấp vào ô nhập liệu mật khẩu
* B7: Nhập mật khẩu
* Nhập mật khẩu là: matkhau12345
* B8: Nhấp vào nút Đăng nhập
* B8: Xác minh đăng nhập thành công bằng cách nhấp chuột vào nút Hồ Sơ
* Sau khi nhấn nút đăng nhập, để kiểm tra xem đã đăng nhập thành công hay chưa thì di chuột đến vị trí hiển trị nút Hồ Sơ và nhấn vào, nếu nhấn được thì đăng nhập thành công.

**D, TC\_04: Bỏ trống cả trường email và mật khẩu**

**Graphical user interface, text, application

Description automatically generated**

* B1: Mở trang web uniqlo.com
* B2: Đặt kích thước cửa sổ (Dẫn đến trang đăng nhập)
* B3: Nhấp chuột vào form đăng nhập
* B4: Nhấp vào nút Đăng nhập
* B5: Xác minh lỗi bằng Khẳng định văn bản (assert text)
* Sau khi nhấn nút đăng nhập, để kiểm tra xem có hiển thị lỗi không thì dùng assert text để kiểm tra. Kiểm tra xem có hiển thị lỗi dạng văn bản là “Vui lòng nhập một địa chỉ email.” Hay không, nếu có là xác minh lỗi thành công.
* B6: Xác minh lỗi bằng Khẳng định văn bản (assert text)
* Tương tự bước 5, sau khi nhấn nút đăng nhập, để kiểm tra xem có hiển thị lỗi không thì dùng assert text để kiểm tra. Kiểm tra xem có hiển thị lỗi dạng văn bản là “Vui lòng nhập mật khẩu.” hay không, nếu có là xác minh lỗi thành công.

## **2.3. Đánh giá các trường hợp thực hiện**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TC** | **Đầu vào** | **Đầu ra** | **Expected output** | **Hệ thống chạy thực tế** | **Đánh giá** |
| TC\_01 | Email: empty  Mật khẩu: matkhau12345 | Từ chối quyền truy cập | Chưa nhập email | Vui lòng nhập một địa chỉ email. | Pass |
| TC\_02 | Email: chuoinguyen895@gmail.com  Mật khẩu: empty | Từ chối quyền truy cập | Chưa nhập mật khẩu | Vui lòng nhập mật khẩu. | Pass |
| TC\_03 | Email: chuoinguyen895@gmail.com  Mật khẩu: matkhau12345 | Cho phép truy cập vào | Chuyển sang trang chủ của uniqlo.com | Chuyển về trang chủ của uniqlo.com | Pass |
| TC\_04 | Email: empty  Mật khẩu: empty | Từ chối quyền truy cập | Chưa nhập email | Vui lòng nhập một địa chỉ email. | Pass |
| Chưa nhập mật khẩu | Vui lòng nhập mật khẩu. |

## **2.4. Mô tả lại đặc tả use case đăng nhập của website bán hàng UNIQLO**

|  |  |
| --- | --- |
| Use Case Name | Đăng nhập |
| Description | Đăng nhập bằng tài khoản xác thực để sử dụng các tính năng được phép mà hệ thống cung cấp. |
| Actor | Người dùng |
| Priority | Must have |
| Trigger | Người dùng truy cập vào đúng website  Thiết bị trong trạng thái kết nối Internet |
| Pre-Condition | Thiết bị sử dụng có kết nối Internet |
| Post-Condition | Người dùng đăng nhập thành công và được thực hiện các chức năng quản lý tài khoản, tìm kiếm và đặt mua và thanh toán hàng hóa. |
| Basic Flow | 1. Tại giao diện trang chủ của Website  2. Người dùng click “Đăng nhập”  3. Hệ thống hiện thị form đăng nhập  4. Người dùng nhập tên đăng nhập và mật khẩu  5. Click đăng nhập  6. Hệ thống hiện thị giao diện tìm kiếm sản phẩm |
| Exception Flow | ***3.a. Người dùng nhập tên đăng nhập nhưng không nhập username***  3.a.1. Hệ thống hiện thị thông báo “Vui lòng nhập mật khẩu”  ***3.b. Người dùng nhập mật khẩu nhưng không dùng tên đăng nhập***  3.b.1. Hệ thống hiện thị thông báo “Vui lòng nhập tên đăng nhập”.  ***3.c. Người dùng không nhập tên đăng nhập và mật khẩu***  3.c.1. Hệ thống hiển thị đồng thời thông báo “Vui lòng nhập tên đăng nhập” và “Vui lòng nhập mật khẩu”.  ***3.d. Người dùng nhập tên đăng nhập hoặc mật khẩu không hợp lệ***  3.d.1. Hệ thống hiển thị “Tài khoản không tồn tại, vui lòng kiểm tra lại”. |

# **PHẦN 3: PHỤ LỤC**

# **1. Chương trình code**

|  |
| --- |
| **Lớp date** |
| #include <iostream>  using namespace std;  class date{  public:  int day, month, year;    // function  void input(){  cout << “Nhap ngay: “; cin >> this->day;  cout << “Nhap thang: “; cin >> this->month;  cout << “Nhap nam: “; cin>> this->year;  }    void display(){  cout << this->day <<”/ “ << this->month <<”/ “ << this->year << endl;  }  }; |

|  |
| --- |
| **Lớp person, employees, customers** |
| #include “date.cpp”  #include <string>  class person{  protected:  string name;  date birth;  int phoneNumber;  int id;  public:  // function  virtual void input(){  cin.ignore();  cout << “Nhap ho ten: “; getline(cin, this->name);  cout << “Nhap ngay sinh: “;  this->birth.input();  cout << “Nhap so dien thoai: “; cin >> this->phoneNumber;  cout << “Nhap so dinh danh ca nhan: “; cin >> this->id;  }    virtual void display(){  cout << “ Ho ten: “ << this->name << endl;  cout << “ Ngay sinh: “;  this->birth.display();  cout << “ So dien thoai: “ << this->phoneNumber <<endl;  cout << “ So dinh danh ca nhan: “ << this->id <<endl;  }  };  class customers: protected person{  public:  void input(){  person :: input();  }  void display(){  person :: display();  }  };  class customers: protected person{  public:  void input(){  person :: input();  }  void display(){  person :: display();  }      }; |

|  |
| --- |
| **Lớp fights và trlFlights** |
| #include <cstdlib>  #include “customers.cpp”  class flights{  private:  int id;  string origin; // diem khoi hanh  string destination; // diem den  float intialPrice;    public:  // function  void input(){  cin.ignore();  cout << “Nhap ma chuyen bay: “; cin >> this->id;  //this->id = rand(); // ma tu sinh ngau nhien  cin.ignore();  cout << “Nhap diem khoi hanh: “; getline(cin, this->origin);  cout << “Nhap diem den: “; getline (cin, this->destination);  cout << “Nhap gia khoi diem: “; cin >> this->intialPrice;  }    void displayCustomers(){  cout << “ “<< this->id << “. “<<this->origin << “ – “ << this->destination << endl;  }    void displayEmployees(){  cout<< “ “ <<this->id << “. “ << this->origin << “ – “ << this->destination << “ “  << this->intialPrice << endl;  }    float getInitialPrice(){  return this->intialPrice;  }  };  class trlFlights{  private:  flights\* listFlight;  int num; // so luong chuyen bay can quan ly  public:  void input(){  cout << “ Nhap so luong thong tin chuyen bay muon them: “; cin >> this->num;  // khoi tao + cap phat vung nho  this->listFlight = new flights [this->num];  for (int I = 0; I < this->num; i++){  cout << endl;  cout << “ Nhap thong tin chuyen bay thu “ << i+1 << endl;  this->listFlight[i].input();  }  cout << “ Them thong tin chuyen bay thanh cong!” << endl << endl;  }  void displayCustomers(){  cout << endl << endl;  cout << “ -----------------------"<<endl;  cout << “ DANH SACH CHUYEN BAY “<< endl;  for (int I = 0; I < this->num; i++){  //cout << endl << endl << “ “ << i+1 <<”. “;  this->listFlight[i].displayCustomers();  }  }  //array[0] cua mang luu thong tin chuyen bay  void displayEmployees(){  cout << endl << endl;  cout << “ ---------------------"<<endl;  cout << “ DANH SACH CHUYEN BAY “<< endl;  for (int I = 0; I < this->num; i++){  //cout << endl << endl << “ “ << i+1 <<”. “;  this->listFlight[i].displayEmployees();  }  }      // ham nay dung de hien thi thong tin chuyen bay ma khach hang chon  int chooseFlights (int choice) {  int I;  for (int I = 0; I < this->num; i++) {  if (I + 1 == choice) {  //return 1;  this->listFlight[i].displayCustomers();  }  }  }    // ham nay dung de lay gia cua chuyen bay  float getPriceInitial(int choice){  int I;  int price;  for (int I = 0; I < this->num; i++) {  if (I + 1 == choice) {  price = this->listFlight[i].getInitialPrice();  }  }  return price;  }  }; |

|  |
| --- |
| **Lớp tickets** |
| #include <iostream>  #include <string>  using namespace std;  class tickets {  private:  int typeTicket; // loai ve {Khu hoi; Mot chieu}  int seatClass; //hang ghe {Binh thuong, thuong gia, hang nhat..}    public:  // loai ve  void inputTypeTicket(){  cout << “ LUA CHON LOAI VE”<< endl;  cout << “ 1. Mot chieu”<< endl;  cout << “ 2. Khu hoi” << endl;  cout << “ Nhap loai ve: “;  cin >> this->typeTicket;    }    // nhap hang ghe  void inputSeatClass(){  cout << “ LUA CHON HANG GHE”<< endl;  cout << “ 1. Pho thong”<< endl;  cout << “ 2. Thuong gia”<< endl;  cout << “ 3. Hang nhat” << endl;  cout << “ Nhap hang ghe: “;  cin >> this->seatClass;  }    void display(){  cout << “Loai ve: “ << this->typeTicket << endl;  cout << “Hang ghe ngoi: “ << this->seatClass;  }    int getTypeTicket(){  return this->typeTicket;  }    int getSeatClass(){  return this->seatClass;  }  }; |

|  |
| --- |
| **Lớp orders** |
| #include “tickets.cpp”  #include “flights.cpp”  #include <cstring>  class orders{  private:  customers c;  flights a;  trlFlights f;  tickets t;  int flightDate; // so ngay dat truoc  float price; // gia ve cuoi cung  //date flightDate; // option 1 -2 -3 🡪 cac khoang doi giong mo ta  float giabienthien2;  float giabienthien3;  float giabienthien4;  float giabienthien5;  float tienve; // gia ve cuoi cung  float initialPrice;  int dtsd; // nhap lua chon cho doi tuong su dung ve  public:        // // lay gia khoi diem cua chuyen bay tuong ung voi lua chon cua khach hang  float calculateInput1(int choice){  this->initialPrice = this->f.getPriceInitial(choice);  return this->initialPrice;  }    // // tinh gia bien thien dua tren input2: ngay khoi hanh  float calculateInput2(){  cout << “ Ngay khoi hanh thuoc [1, 30]: “ << endl;  cout << “ Nhap so ngay ban muon dat ve truoc: “; cin >> this->flightDate;  while (this->flightDate<1 || this->flightDate>30 || cin.fail()){  cout << “ Nhap ngay khoi hanh khong hop le!” << endl;  cout << “ Vui long nhap lai so ngay ban muon dat ve truoc: “;  cin.clear(); // Xoa cac bit loi trong buffer  cin.ignore(10000, ‘\n’); // Loai bo toan bo du lieu nhap tu ban phim  cin >> this->flightDate;  }  if (this->flightDate <=7){  return this->giabienthien2 = this->initialPrice\* 0.15;  }  else if (this->flightDate <=15){  return this->giabienthien2 = 0;  }  else {  return this->giabienthien2 = - this->initialPrice \* 0.15;  }  }    // tinh gia bien thien dua tren input3: hang ghe  float calculateInput3(){  this->t.inputSeatClass();  if (this->t.getSeatClass() == 1)  return this->giabienthien3 = 0;  if (this->t.getSeatClass() == 2)  return this->giabienthien3 = this->initialPrice \* 0.2;  if (this->t.getSeatClass() == 3)  return this->giabienthien3 = this->initialPrice \* 0.4;  }    // convert bien hang ghe dang so ve dang chu  string convertSeatClass(){  if (this->t.getSeatClass() == 1)  return “Pho thong”;  if (this->t.getSeatClass() == 2)  return “Thuong gia”;  if (this->t.getSeatClass() == 3)  return “Hang nhat”;  }    float calculateInput4(){  cout << “ LUA CHON DOI TUONG SU DUNG VE”<< endl;  cout << “ Nguoi lon – 13 tuoi tro len”<< endl;  cout << “ Tre em – (0-12 tuoi)”<< endl;  cout << “ Quy dinh doi tuong su dung may bay: (0 – 100] – tuoi” << endl;  cout << “ Nhap tuoi cua doi tuong su dung ve: “;  cin >> this->dtsd;  while (this->dtsd<1 || this->dtsd>100 || cin.fail()){  cout << “ Nhap tuoi khong hop le!” << endl;  cout << “ Vui long nhap lai so so tuoi: “;  cin.clear(); // Xoa cac bit loi trong buffer  cin.ignore(10000, ‘\n’); // Loai bo toan bo du lieu nhap tu ban phim  cin >> this->dtsd;  }    if (this->dtsd <=12){    return this->giabienthien4 = -this->initialPrice\*0.1;  }  else {  return this->giabienthien4 = 0;  }  }    string convertAge(){  if (this->dtsd> 0 && this->dtsd <=12){    return “Tre em”;  }  if (this->dtsd >12 && this->dtsd <=100){  return “Nguoi lon”;  }    }    // tinh gia bien thien dua tren input4: loai ve: khu hoi, 1 chieu  float calculateInput5(){  t.inputTypeTicket();  //int temp = this->calculateInput1(choice);  if (this->t.getTypeTicket() == 1){    return this->giabienthien5 = 0;  }  if (this->t.getTypeTicket() == 2) {  return this->giabienthien5 = this->initialPrice \* 0.8;  }  }    // chuyen doi lua chon (1,2 dang so) sang chu  string convertTypeTicket(){  if (this->t.getTypeTicket() == 1)  return “Mot chieu”;  if (this->t.getTypeTicket() == 2)  return “Khu hoi”;  }  //  int calculatePriceTicket(){    this->tienve = this->initialPrice + this->giabienthien2 + this->giabienthien3 + this->giabienthien4 + this->giabienthien5;  return this->tienve;  }  // hien thi thong tin ve ma khach chon  void detailTicket(int choice){  cout << “ THONG TIN VE NHU SAU: “<< endl << endl << endl;  cout << “ “; f.chooseFlights(choice); cout<<endl;  cout << “ Dat truoc: “ << this->flightDate << “- ngay” << endl;  cout << “ Hang ghe: “ << this->convertSeatClass() <<endl;  cout << “ Doi tuong su dung: “ << this->convertAge()<<endl;  cout << “ Loai ve: “ << this->convertTypeTicket() <<endl;  cout << “ Gia ve: “ << this->calculatePriceTicket() << endl;  }      // tuong duong voi menu khach hang  void placeAnOrder(){  this->f.displayCustomers();  int choice;  cout << “Chon chuyen bay: “; cin >> choice;  f.chooseFlights(choice);  this->calculateInput1(choice);  cout << this->calculateInput2();  //cout << “Gia bien thien 2: “ << this->calculateInput2(this->flightDate, this->initialPrice, this->giabienthien2) << endl;  cout << “Gia bien thien 3: “ << this->calculateInput3() << endl;  cout << “Gia bien thien 4: “ << this->calculateInput4() << endl;  cout << “Gia bien thien 5: “ << this->calculateInput5() << endl;  cout << “ --------------------------" << endl;  this->detailTicket(choice);  while (true){  cout << “ ----------------------"<< endl;  cout << “ GIAO DIEN DAT VE” << endl << endl;  cout << “ 1. Dat ve” << endl;  cout << “ 0. Thoat!”<< endl;  cout << “ ----------------------" << endl << endl;  cout << “ Nhap lua chon: “;    int choicePlaceAnOrder;  cin >> choicePlaceAnOrder;  if (choicePlaceAnOrder == 1){  this->c.input();    this->detailTicket(choice);  cout << “ THONG TIN KHACH HANG NHU SAU:” << endl;  this->c.display();  cout << “ DAT VE THANH CONG ! “ << endl << endl;  return ;  }      else if (choicePlaceAnOrder == 0){  return;  }  else {  cout << “ Lua chon khong hop le!” << endl;  }  }  }        void menuEmployee(){  while (true){  cout << “ ------------------------" << endl;  cout << “ GIAO DIEN NHAN VIEN” << endl << endl;  cout << “ 1. Them chuyen bay” << endl;  cout << “ 2. Xem thong tin chuyen bay” << endl;  cout << “ 0. Thoat” << endl << endl << endl;  cout << “ Nhap lua chon: “;    int choiceEmployee;  cin >> choiceEmployee;    if (choiceEmployee == 1){  this->f.input();  }  else if (choiceEmployee == 2){  this->f.displayEmployees();  }  else if (choiceEmployee == 0){  return ;  }  else {  cout << “ Lua chon khong hop le!” << endl;  }  }  }    }; |

|  |
| --- |
| **Hàm main** |
| #include “orders.cpp”  int main()  {  while (true){  cout << “ ------------------------" << endl;  cout << “ QUAN LY DAT VE” << endl << endl << endl;  cout << “ 1. Vai tro nhan vien” << endl;  cout << “ 2. Vai tro khach hang” << endl;  cout << “ 0. Thoat” << endl << endl << endl;  cout << “ Nhap lua chon: “;    int choice;  cin >> choice;  orders o;  if (choice == 1){  string username;  int password;  do {  cout << endl << endl << endl;  cout << “ GIAO DIEN DANG NHAP” <<endl << endl << endl;  cout << “ Nhap Username Va Password” << endl;  cout << “ Username: “; cin >> username;  cout << “ Passrord: “; cin >> password;  } while (username != “group3” && password != 3);  o.menuEmployee();  }    else if (choice == 2){  while (true)  {  int action;  cout << “ ------------------------" << endl;  cout << “ VAI TRO KHACH HANG” << endl << endl << endl;  cout << “ 1. Lua chon ve” << endl;  cout << “ 0. Thoat” << endl << endl << endl;  cout << “ Nhap lua chon: “; cin>> action;  if (action ==1)  o.placeAnOrder();  else if (action == 0){  return 0;  }    }  }  else if (choice == 0){  return 0;  }  else {  cout << “ Nhap khong hop le!” << endl;  }    }    }; |

# **2. Biên bản thảo luận**

**BIÊN BẢN HỌP NHÓM LẦN 1:**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

**----o0o----**

**BIÊN BẢN LÀM VIỆC NHÓM**

*(Lần 1)*

Bộ môn: Kiểm thử phần mềm

Giảng viên hướng dẫn: Vũ Diệu Hương

Nhóm: 03

Nhóm trưởng: Ngô Thị Hoài

Lớp học phần: 2305ECIT3021

Thời gian làm việc: từ 20h00 đến 22h30, ngày 22 tháng 03 năm 2023

Địa điểm: Google Meet

Thành viên có mặt: 9/9

Vắng: 0

Mục tiêu cuộc họp:

1. Lựa chọn đề tài thảo luận

2. Tìm hiểu, phác thảo đề cương, phân tích đề tài thảo luận và tìm ra phương hướng,

giải pháp giải quyết đề tài thảo luận

3. Phân công cơ bản công việc cần làm và thời gian thực hiện cụ thể cho từng công

việc được giao

Đánh giá chung kết quả cuộc họp: Tốt. Các thành viên đều hoàn thành tốt nhiệm vụ được phân công, nhiệt tình làm các công việc khác khi được nhóm trưởng giao nhiệm vụ.

Hà Nội, ngày 22 tháng 3 năm 2023

Người lập biên bản

Khuyên

Lê Thị Khuyên

Nhóm trưởng

Hoài

Ngô Thị Hoài

**BIÊN BẢN HỌP NHÓM LẦN 2:**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

**----o0o----**

**BIÊN BẢN LÀM VIỆC NHÓM**

*(Lần 2)*

Bộ môn: Kiểm thử phần mềm

Giảng viên hướng dẫn: Vũ Diệu Hương

Nhóm: 03

Nhóm trưởng: Ngô Thị Hoài

Lớp học phần: 2305ECIT3021

Thời gian làm việc: từ 21h00 đến 23h00, ngày 02 tháng 04 năm 2023

Địa điểm: Google Meet

Thành viên có mặt: 9/9

Mục tiêu cuộc họp:

1. Trao đổi lại về bài toán sau buổi thuyết trình

2. Phân công lại công việc cần làm và thời gian thực hiện cụ thể cho từng công việc được giao

Đánh giá chung kết quả cuộc họp: Tốt. Các thành viên đều tham gia đóng góp ý kiến để hoàn thiện bài thảo luận.

Hà Nội, ngày 02 tháng 04 năm 2023

Nhóm trưởng

Người lập biên bản

Khuyên

Lê Thị Khuyên

Hoài

Ngô Thị Hoài

**BIÊN BẢN HỌP NHÓM LẦN 3:**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

**----o0o----**

**BIÊN BẢN LÀM VIỆC NHÓM**

*(Lần 3)*

Bộ môn: Kiểm thử phần mềm

Giảng viên hướng dẫn: Vũ Diệu Hương

Nhóm: 03

Nhóm trưởng: Ngô Thị Hoài

Lớp học phần: 2305ECIT3021

Thời gian làm việc: từ 20h00 đến 21h30, ngày 25 tháng 04 năm 2023

Địa điểm: Google Meet

Thành viên có mặt: 9/9

Vắng: 0

Mục tiêu cuộc họp:

1. Hoàn thiện Word

2. Thuyết trình kết hợp Powerpoint

3. Nhóm trưởng đánh giá mức độ tham gia và hoàn thành công việc của từng thành viên

Đánh giá chung kết quả cuộc họp: Tốt. Các thành viên đều hoàn thành tốt nhiệm vụ được phân công, nhiệt tình làm các công việc khác khi được nhóm trưởng giao nhiệm vụ.

Hà Nội, ngày 25 tháng 04 năm 2023

Nhóm trưởng

Người lập biên bản

Khuyên

Lê Thị Khuyên

Hoài

Ngô Thị Hoài

# **KẾT LUẬN**

Kiểm thử phần mềm nói chung và kiểm thử tự động bằng Senenium trên nền Website nói riêng là một vấn đề hết sức quan trọng đối với các tổ chức phát triển phần mềm hiện nay. Trong quá trình thực hiện bài thảo luận của mình do thời gian nghiên cứu và kiến thức của nhóm còn nhiều hạn chế nên một số phần của bài thảo luận còn chưa được sâu, còn nhiều thiếu sót. Nhóm 3 rất mong nhận được sự nhận xét, đánh giá và góp ý từ cô cũng như các bạn trong lớp để nhóm có thể hoàn thiện bài thảo luận này một cách tốt hơn.

Một lần nữa, chúng em xin cảm ơn TS. Vũ Diệu Hương đã hướng dẫn nhóm em thực hiện bài thảo luận này.